



TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA
ELEKTRONIIKAN JA TIETOLIIKENNETEKNIIKAN TUTKINTO-OHJELMA

DIPLOMITYÖ

RFIC PIIRIN DIGITAALIOHJAUKSEEN VAADITTAVIEN PIIRIRAKENTEIDEN SUUNNITTELU

Tekijä

Veeti Kiuru

Valvoja

Janne Aikio

Toinen tarkastaja

Timo Rahkonen

Työn tekninen ohjaaja

Janne Aikio

Elokuu 2020

Kiuru V. (2020) RFIC piirin digitaaliohjaukseen vaadittavien piirirakenteiden suunnittelu. Oulun yliopisto, tieto- ja sähkötekniikan tiedekunta, elektroniikan ja tietoliikennetekniikan tutkinto-ohjelma. Diplomityö, 71 s.

TIIVISTELMÄ

Tässä työssä suunnitellaan 22 nm FD-SOI-prosessille digitaalikirjaston osa, jolla voidaan toteuttaa tarvittava ohjaus RFIC-komponenttilohkoille. Komponenteista rakennetaan mahdollisimman pieniä ja nopeita käyttäen matalan kynnyksjännitteen transistoreja. Työssä tarvittavan digitaalisen ohjauksen luomiseksi tehdään NAND-, NOR-, AND- ja NOT-portit sekä siirtorekistereitä, osoitetekooderi, virta-DAC ja D-kiikku.

Piirikaaviot ja -kuviot suunnitellaan Cadencen Virtuoso-ohjelmistolla ja simuloinnit tehdään Cadencen ADE-simulointiympäristössä. Simulointien avulla mitoitetaan digitaalilohkot ja varmistetaan niiden toiminta. Testipenkkeihin luodaan lohkoille mahdollisimman todenmukainen ympäristö käyttäen viivästettyjä ja noin 5 ps nousu- ja laskuajoilla olevia signaaleja sekä kuormana 5 minimikokoista invertteriä. Tehonkulutus ja pinta-ala minimoidaan suunnittelemalla mahdollisimman nopeita ja pieniä piirejä ja käyttäen vain muutamaa alinta metallikerrosta piirikuviossa.

Transistoreja yhdistelemällä piirin pinta-alaa saadaan parhaissa tapauksissa pienennettyä yli 50%. Simuloinneissa nähdään, että komponenttien tehonkulutus ja teknologian tuomat virtarajoitukset eivät tuota ongelmia, sillä komponenttien maksimivirrat pysyvät alle 500 μA . Lisäksi simuloinneissa tulee esille logiikkakomponenttien kyky vaimentaa kohinaa ja kapasitiivisen kuormituksen vaikutus virtaan ja viiveeseen. Lopuksi valmiit komponentit simuloidaan vielä RFIC-lohkon kanssa piirikaaviotasolla ekstraktoitujen parasiittisten vaikutus huomioiden.

Avainsanat: RFIC, Cadence, digitaalikirjasto, parasiittiset komponentit, matala kynnysjännite, transistori, tehonkulutus.

Kiuru V. (2020) Design of Circuit Blocks For Digital Control of Integrated RF Circuit.
University of Oulu, Degree Programme in Electronics and Communications Engineering.
Master's Thesis, 71 p.

ABSTRACT

The aim of this work is to design a part of a digital component library using 22 nm FD-SOI CMOS process, which could be used to digitally control analog RFIC-blocks. Low threshold voltage transistors are used in order to create as small and fast components as possible. In this work NAND, NOR, AND and NOT logic ports are designed together with shift registers, address decoder, current-DAC and D-flipflop.

Schematics and layouts are designed by using Cadence Virtuoso software and simulations are done by using Cadence ADE -simulation environment. Simulations are used to size logic components and verify operations. A realistic operation environment is created by using delayed signals with 5 ps rise and fall times and a load of 5 minimum size inverters. Power consumption and area of circuits are minimized by designing fast and small circuits and by using only a few of the bottom metal layers in layouts.

In the best cases the area is reduced more than 50 percent by combining transistor structures. In simulations power consumption and current constraints imposed by used technology are not a problem due to smaller than 500 μA maximum currents of components. In addition, the simulations show the ability of logic components to attenuate noise and how the capacitive load affects to the current consumption and delay of components. Finally, implemented components are simulated with actual RFIC-blocks at schematic level by considering the effects of extracted parasitic components.

Key words: RFIC, Cadence, Digital component library, parasitic component model, low threshold voltage, power consumption.

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUETTELO

ALKULAUSE

LYHENTEIDEN JA MERKKIEN SELITYKSET

1	JOHDANTO	9
2	DIGITAALISUUNNITTELU	10
2.1	Digitaalilogiikka	10
2.2	Tehonkulutus	12
2.2.1	Dynaaminen tehonkulutus	12
2.2.2	Staatinen tehonkulutus	14
2.2.3	Tehonkulutuksen vaikutuksia suunnitteluun	15
2.3	Kohinamarginaalit	15
2.4	Komponenttikirjastot	16
2.4.1	Prosessi-, jännite- ja lämpötilavaihtelut	16
2.4.2	Vuotovirta- ja kynnysjännite	17
2.5	FD-SOI Prosessi	18
3	PIIRIKAAVIOIDEN SUUNNITTELU	19
3.1	Työn tavoitteet ja tarvittavat komponentit	19
3.2	Peruskomponentit ja niiden transistorimallit	20
3.2.1	NAND- ja NOR-portti	20
3.2.2	Invertteri ja puskuri	21
3.2.3	Multiplekseri, D-salpa ja Schmitt-triggeri	22
3.3	Peruskomponenttien hyödyntäminen monimutkaisemmissa rakenteissa	24
3.3.1	AND-portti, siirtorekisteri ja osoitdekooderi	25
3.3.2	D-Kiikku	27
3.3.3	4-tuloinen Multiplekseri BIN/LIN dekodauksella	28
3.3.4	Virta-DAC	30
4	PIIRIRAKENTEIDEN SIMULOINTI	32
4.1	Rakenteiden toiminta	32
4.2	Kuorma ja toimintaympäristö	33
4.2.1	Fan-in ja Fan-out	34
4.2.2	Kapasitanssit, resistanssit ja viive	35
4.3	Testipenkit ja simulointi	37
4.4	D-Kiikun simulointitulokset	39
4.4.1	Viiveen, setup- ja pito-ajan mittaukset	39
4.4.2	Prosessi- ja lämpötilavaihteluiden vaikutus viiveeseen	41
4.5	4-tuloinen multiplekserin simulointitulokset	42
4.6	Virta DAC simulointitulokset	43
4.6.1	Virta DAC jännitealue	45
4.6.2	Virta DAC jännitealueen linearisointi	45

4.7	Logiikan ominaisuuksien simulointia	48
4.7.1	Virta ja tehonkulutuksen mittaaminen	48
4.7.2	Kuorman vaikutuksia toimintaan	49
4.7.3	Kohinan vaikutuksia toimintaan	51
5	PIIRIKUVIOIDEN SUUNNITTELU	53
5.1	Suunnittelun periaatteet ja tavoitteet	53
5.1.1	Piirikuvioiden koko	54
5.2	Rakenteiden tekeminen	55
5.3	Calibre nmLVS tarkistus	57
5.4	Calibre nmDRC tarkistus	58
5.5	Parasiittisten ekstraktointi Calibre xACT:n avulla	58
5.6	Toiminnallisuuden varmentaminen ekstraktoitujen rakenteiden avulla	59
5.7	Piirirakenteiden simulointi varsinaisten lohkojen kanssa	61
6	POHDINTA	66
7	YHTEENVETO	68
8	LÄHDELUETTELO	69
9	LIITTELUETTELO	71

ALKULAUSE

Tämä työ on tehty osana MIMEPA -tutkimusprojektia Oulun yliopistolla. Tarkastajana ja valvojana työlle toimi dosentti Janne Aikio ja toisena tarkastajana professori Timo Rahkonen. Haluan kiittää heitä molempia työn tarkastamisesta ja hyvistä näkökulmista ja ohjauksesta työn aikana. Janne ja Timo rohkaisivat minua suunnittelemaan ja rakentamaan työssä tarvittavat komponentit todella vapaalla kädellä, mutta samalla ammattitaitoisesti. Haluan myös tuoda kiitoksen Jere Rusaselle ja Alok Sethille, jotka ystävällisesti auttoivat minua saamaan osan tarvittavista simuloinneista tehtyä, sekä auttoivat myös projektin etenemisessä.

Erityiset kiitokset haluan tuoda omalle perheelleni ja vaimolleni Reetta Kiurulle suuresta tuesta ja kannustuksesta koko yliopisto-opintojen ajan sekä kavereilleni, jotka ovat kannustaneet opiskelemaan ja opiskelleet yhdessä minun kanssani.

Diplomityön aihepiiri oli mielenkiintoinen ja tarjosi loistavat mahdollisuudet oppia käyttämään piirisuunnitteluohjelmistoa, sekä tarjosi todella kattavan tietopaketin digitaalilogiikkakomponenteista sekä niiden suunnittelusta.

Oulussa 12.08.2020

Veeti Kiuru

LYHENTEIDEN JA MERKKIEN SELITYKSET

BIN/LIN	Binary-Line
BOX	Buried Oxide, haudattu oksidi
CDAC	Current Digital-to-Analog Converter, virta-digitaali-analogiamuunnin
DAC	Digital-to-Analog Converter, digitaali-analogiamuunnin
DRC	Design Rules Check, suunnittelusääntöjen tarkistus
DTL	Diode-Transistor Logic, diodi-transistori logiikka
ECL	Emitter-Coupled Logic, emitterikytketty logiikka
FBB	Forward Body Bias, myötäsuuntainen rungon biasjännite
FF	Fast-Fast, nopea-nopea
FD-SOI	Fully Depleted Silicon On Insulator, täysin tyhjennetty piieristerakenne
FO4	Fanout of Four, neljän portin fanout
FOM	Figure of Merit, hyvyysluku
GND	Ground, maataso
HB	Harmonic Balance, harmoninen balanssi
HVT	High V_T , korkea V_T
LVS	Layout Versus Schematic, piirikuvio verrattuna piirikaavioon
LVT	Low V_T , matala V_T
MOM	Metal-Oxide-Metal, metalli-oksidi-metalli
MOS	Metal-Oxide-Semiconductor, metalli-oksidi-puolijohde
PA	Power Amplifier, tehovahvistin
RBB	Reverse Body Bias, käänteinen rungon biasjännite
RF	Radio Frequency, radiotaajuus
RFIC	Radio Frequency Integrated Circuit, radiotaajuinen integroitu piiri
RTL	Resistor-Transistor Logic, vastus-transistori logiikka
RVT	Regular V_T , normaali V_T
SLVT	Super Low V_T , todella matala V_T
SS	Slow-Slow, hidas-hidas
TT	Typical-Typical, tyypillinen-tyypillinen
TTL	Transistor-Transistor Logic, transistori-transistori logiikka
UHVT	Ultra high V_T , todella korkea V_T
VDD	Positive supply voltage, positiivinen käyttöjännite
VLSI	Very Large Scale Integration, todella suuren mittakaavan integrointi
VSS	Negative supply voltage, negatiivinen käyttöjännite

C_L	Kuormakapasitanssi
C_{ox}	Hilaoksidin kapasitanssi
e	Neperin luku
E	Käyttöjännitteestä otettu energia
E_{C_L}	Kuormakapasitanssiin ladattu energia
f	Taajuus
$i(t)$	Hetkellinen virta
I	Virta
I_D	Nieluvirta
I_{off}	Johtamattomassa tilassa olevan transistorin virta
I_{on}	Johtavassa tilassa olevan transistorin virta
K	Transistorien koosta ja käytetystä teknologiasta riippuva vakio

L	Transistorin pituus
NM_H	Kohinamarginaali, korkea
NM_L	Kohinamarginaali, matala
$p(t)$	Hetkellinen teho
$P_{kytkentä}$	Kytkentäteho
$P_{oikosulku}$	Oikosulkuteho
R	Resistanssi
R_{off}	Kiinni-resistanssi
R_{on}	Päällä-resistanssi
S	Transistorin leveyden ja pituuden suhde
t	Aika
U_{off}	Johtamattomassa tilassa olevan transistorin jännitehäviö
U_{on}	Johtavassa tilassa olevan transistorin jännitehäviö
$v(t)$	Hetkellinen jännite
V_{BG}	Back-gate jännite
V_{GS}	Transistorin hila-lähdejännite
V_0	Lähtöjännite
V_{DD}	Positiivinen käyttöjännite
V_{IHmin}	Tulon korkeamman potentiaalin minimi
V_{ILmax}	Tulon matalamman potentiaalin maksimi
V_{OHmin}	Lähdön korkeamman potentiaalin minimi
V_{OLmax}	Lähdön matalamman potentiaalin maksimi
V_{th}	Kynnysjännite
V_{tn}	NMOS transistorin kynnysjännite
W	Transistorin leveys
τ	Aikavakio
μ_n	NMOS transistorin varausten liikkuvuus
\ln	Luonnollinen logaritmi
$\frac{d}{dt}$	Derivaatta ajan suhteen
$\int_0^t dt$	Integraali tietyn jakson t yli

1 JOHDANTO

Tämän diplomityön aiheena on RFIC-piirin jänniteohjaukseen, sekä analogisten kytkimien digitaalisen ohjaukseen soveltuvan kustomoidun logiikkakomponenttien kokoelman rakentaminen. Logiikkakomponentit vaihtavat tilaansa bittiohjauksen funktiona ja niiden avulla voidaan rakentaa isoja toiminnallisia yksiköitä, kuten esimerkiksi rekistereitä ja kontrollereita. Logiikkakomponenteista luodaan piirikaaviot ja piirikuviot, sekä tarvittavat mallit, jotta toiminnasta saadaan kattava kuva.

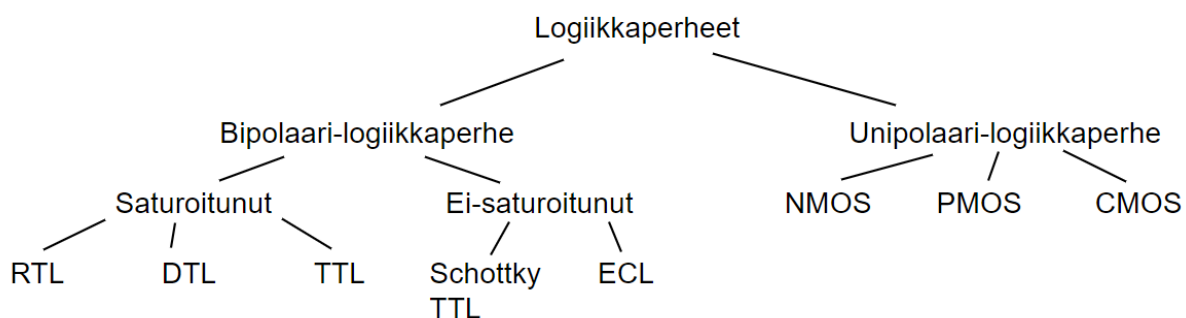
Digitaaliohjauksen määrä integroiduissa piireissä kasvaa jatkuvasti, koska uudet IC-teknologiat mahdollistavat pienempien rakenteiden ja pienemmän tehonkulutuksen suunnittelun. Käyttötaajuuksien kasvaminen pakottaa ohjauslogiikan toimintaa koko ajan nopeammaksi. Nanometriteknologiat skaalaavat myös käyttöjännitteitä piireillä alaspäin, sillä pienet rakenteet eivät kestä suuria virtoja ja jännitteitä. Teknologian muuttuessa logiikka itsessään ei kuitenkaan muutu, vaan peruskomponenttien toimintaperiaate pysyy samanlaisena. Tässä kuitenkin lähestytään rajaa, sillä käyttöjännitteiden madaltuessa noin 150 mV alapuolelle ei logiikkaportteista saa enää vahvistusta, jolloin aikaisempien asteiden kohina vuotaa vaimentumattomana läpi ja regenerointi katoaa kokonaan. Tällöin joudutaan suunnittelemaan virheenkorjaavaa digitaalilogiikkaa. [1] [2] [3]

Analogisia piirejä ohjataan useimmiten virran tai jännitteen avulla. Digitaalinen ohjaus on mahdollista toteuttaa siten, että logiikkakomponentit säätävät halutun jännitteen tai virran ohjattavaan kohteeseen esimerkiksi kytkemällä virtapeilejä käyttöön. Digitaalisten piirien pieni koko mahdollistaa sen, että niitä voidaan sijoittaa hyvin paljon samaan rakenteeseen esimerkiksi RFIC-komponenttien kanssa. Useimmiten analogiakomponenteissa on isoja kondensaattoreita ja keloja, jotka vievät ison osan pinta-alasta ja rakenteiden väliin jää tyhjää tilaa. Lisäämällä piirille digitaaliohjauslohkoja voidaan vähentää kontakteja ulkomailmaan ja saada toiminnasta hyvin ohjelmoitavaa.

Tämä diplomityö koostuu neljästä kokonaisuudesta. Ensimmäisessä osassa käsitellään logiikkaperheisiin ja logiikkaportteihin liittyvää teoriaa, jonka pohjalta perustellaan työssä kehitettävien piirien suunnitteluperiaatteet. Toisena käydään läpi tarvittavat komponentit rakenteineen sekä mitoitus ja toimintaperiaatteet. Kolmannessa osiossa käydään läpi simulointiympäristöä ja mitoitusprosessia, sekä tarkastellaan komponenttien todellista toimintaympäristöä suunnittelun näkökulmasta. Tässä kokonaisuudessa käydään myös läpi simulointituloksia kuten viivemittauksia, kapasitiivisen kuormituksen vaikutuksia toimintaan ja tehonkulutusta. Viimeinen kokonaisuus sisältää piirikuvioiden suunnitteluun liittyviä asioita, kuten käytetyt rakenteet, johdotukset ja rakenteiden suunnitteluprosessin. Osio myös sisältää piirikuvioiden varmennuksen ja parasiittisten ekstraktoinnin sekä komponenttien hyödyntämisen varsinaisten RFIC-lohkojen ohjaukseen RF-simuloinneissa.

2 DIGITAALISUUNNITTELU

Digitaalinen IC-piiri (Integrated Circuit) on pii-puolijohdekide, jonka päälle on rakennettu elektronisista komponenteista digitaalilogiikkaportteja. Digitaalisista IC-piireistä puhuttaessa on tärkeää tuntea logiikan taustalla olevat kytkennät eli perustuvatko portit esimerkiksi bibolaaritransistori kytkentöihin vai CMOS (Complementary Metal Oxide Semiconductor) transistorien hyväksikäyttöön. Kuvassa 1 on esitetty logiikkaperheiden jakautumiskaavio, jossa on useita eri, osin historiallisia, logiikkaperheitä, kuten RTL (Resistor-Transistor Logic), TTL (Transistor-Transistor Logic) ja ECL (Emitter-Coupled Logic). Tässä työssä käytettiin staattista CMOS-logiikkaa ja modernia 22 nm FD-SOI -prosessia. Digitaaliset IC-piirit voidaan kategorisoida myös muilla tavoilla. Logiikkaperheiden lisäksi puhutaan usein integrointiasteesta, eli siitä kuinka paljon logiikkaportteja on yhdellä IC-piirillä. SSI (Small-scale Integration), MSI (Medium-scale Integration), LSI (Large-scale Integration) ja VLSI (Very Large-scale Integration) kertovat, kuinka paljon logiikkaportteja digitaalisella IC-piirillä on. SSI sisältää muutamia portteja, joiden tulot ja lähdöt on suoraan kytketty pakkauksen pinneihin, kun taas VLSI voi sisältää jopa miljoonia portteja yhdellä IC:llä. [2]



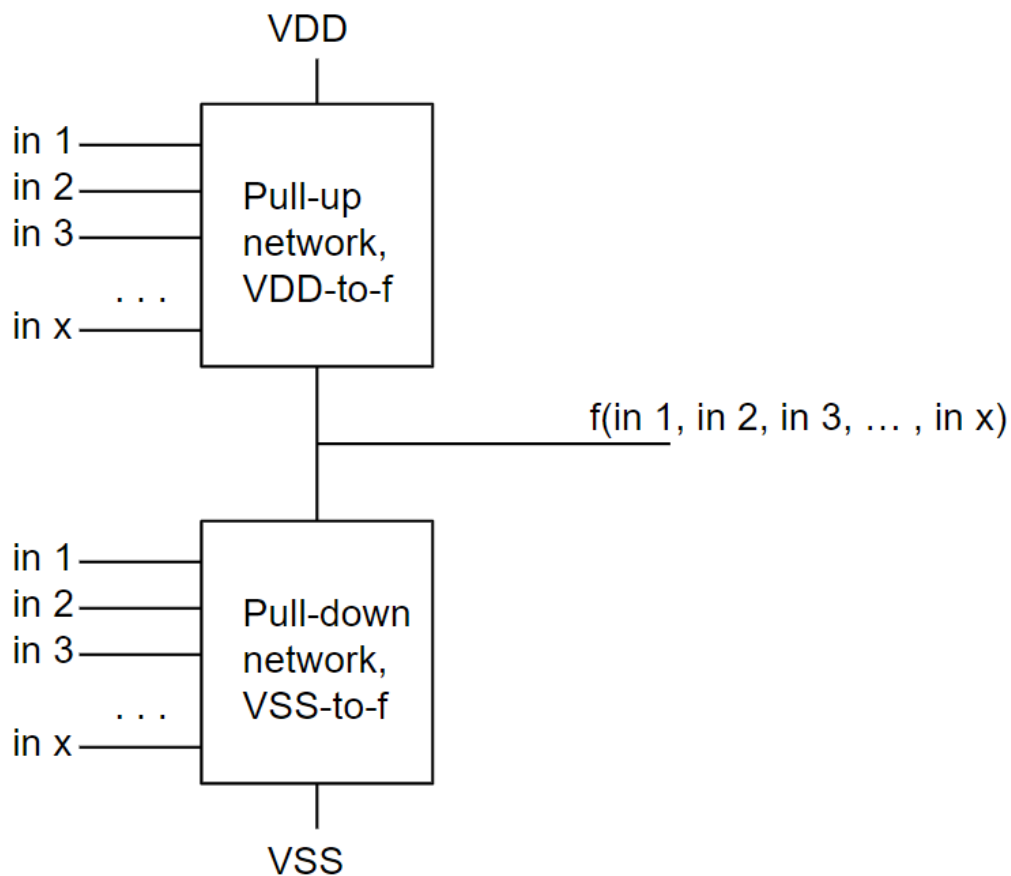
Kuva 1. Logiikkaperhekaavio. RTL = Resistor-Transistor Logic, DTL = Diode-Transistor Logic, TTL = Transistor-Transistor Logic, ECL = Emitter Coupled Logic, NMOS, PMOS ja CMOS = N-channel, P-channel ja Complementary Metal Oxide Semiconductor.

2.1 Digitaalilogiikka

Digitaalilogiikassa on kaksi jännitetasoa, looginen 0 (yleensä 0 V eli maataso) ja looginen 1 (yleensä VDD eli käyttöjännite). Logiikkakomponentin jokaisessa tuloportissa on toinen näistä logiikkatasoista ja tuloporttien jännitetasot määräävät sen, mitkä transistorit ovat komponentin sisällä johtavana ja mitkä eivät johda, jolloin lähtöön saadaan tulojen funktiona tietty logiikkataso. Nimitys logiikkakomponentti tulee siitä, että logiikkaportti tekee tulojen perusteella ”päätöksen” ja kytkee lähdön joko matalaan tai korkeaan potentiaaliin eli tulojen ja lähdön välillä on looginen yhteys. Esimerkiksi AND-portti, jonka kaikkien tulosignaalien ollessa korkeammassa potentiaalissa siirtyy lähtö myös korkeampaan potentiaaliin. [2]

Useimmiten CMOS-logiikkakomponentti koostuu kuvan 2 mukaisesta p- ja n-tyyppin ylös- ja alasveto transistorilohkoista. Ylempi lohko kytkee lähdön käyttöjännitteeseen aina kun portin on tarkoitus nostaa lähtö korkeaan potentiaaliin ja alempi lohko kytkee lähdön maahan aina kun sen on tarkoitus olla matalassa potentiaalissa. Tulosignaali ohjaavat näitä ylös- ja alasveto lohkoja. Ylösvetolohko voi olla myös puhtaasti resistiivinen eli käytännössä ylösvetovastus. CMOS-logiikka on yleensä kääntävää, mikä tarkoittaa sitä, että tulosignaalin ollessa 0, ylösvetolohko kytkeytyy ja tulon ollessa 1, alasvetolohko kytkeytyy eli lähtö signaali on käänteinen tulon nähden. Tämä johtuu siitä, että p-tyyppin transistorit johtavat, kun hilajännite

on matala ja n-tyypin transistorit johtavat, kun hilajännite on korkea. Logiikassa transistorit ajatellaan usein kytkiminä eli johtavana ne vain yhdistävät kaksi paikkaa toisiinsa. Tällainen ajattelutapa helpottaa logiikan toiminnan analysointia ja suunnittelua. [3] [4]



Kuva 2. Esimerkki logiikkaportista ylös- ja alasvetolohkojen avulla esitettynä.

Logiikkaperhe sisältää ison määrän erilaisia logiikkakomponentteja, peruskomponenteista aina isompiin kokonaisuuksiin asti. Peruskomponentteja ovat sellaiset komponentit, joista logiikkaperhe sisältää transistoritason mallit ja joista voidaan rakentaa isompia kokonaisuuksia ja uutta logiikkaa kytkemällä niitä yhteen halutulla tavalla. Jokainen logiikkaperhe sisältää NAND-, NOR- ja NOT-portit, jotka ovat hyvin yleisiä rakennusosia monimutkaisemmissa piireissä. Lisäksi logiikkaperheistä löytyy datakirja, joka listaa kaikki komponentit kyseisessä logiikkaperheessä [2]. Taulukoissa 1 a)-c) on esitetty perusporttien totuustaulut, joista ilmenee niiden toiminta. NAND-portin lähtö on 0, jos molemmat tulosignaalit ovat 1, muulloin lähtö on 1. NOR-portin lähtö on 1, jos molemmat tulosignaalit ovat 0, muulloin lähtö on 0. NOT-portti eli invertteri kääntää signaalitason ympäri eli jos tulossa on 0, lähdössä on 1 ja päinvastoin.

Logiikkaperheiden tärkeitä ominaisuuksia eli FOM (Figure of Merit) ovat esimerkiksi nopeus, fan-in, fan-out (kts kpl 4.2.1), tehonkulutus, viive ja kohinarajat. Muita kiinnostavia asioita analysoinnissa ovat etenemisviiveen ja tehonkulutuksen tulo, maksimi kohina ilman lähtöjännitteen muutosta ja erilaisten toimintojen määrä logiikkaperheessä. Tässä työssä arvioidaan komponentteja tehonkulutuksen, kuormituksen, viiveen ja kohinan avulla.

Taulukko 1. NAND-, NOR- ja NOT-porttien totuustaulut [2].

a)

NAND		
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

b)

NOR		
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

c)

NOT	
A	Y
0	1
1	0

2.2 Tehonkulutus

Kun valmistusprosessit ovat mahdollistaneet alle 100 nm CMOS teknologiat, on samalla myös niihin liittyviä haasteita tullut enemmän [5]. Digitaalilogiikassa tehonkulutus on tärkeä osa suunnittelua. Iso osa tehonkulutuksesta digitaalilogiikassa on yleensä tullut dynaamisen tehonkulutuksen kautta, mutta pienenevät rakenteet ja alemmas skaalautuvat käyttö- ja kynnysjännitteet ovat nostaneet staattisen tehonkulutuksen merkittäväksi teemaksi CMOS transistoreja hyödyntävässä digitaalisuunnittelussa. Staattisen tehonkulutuksen kasvu näkyy niin aktiivisessa- kuin valmiustilassa olevien lohkojen tehonkulutuksessa [6, Fig. 25]. Tämä on aiheuttanut myös sen, että on kehitetty monenlaisia tapoja optimoida tehonkulutusta.

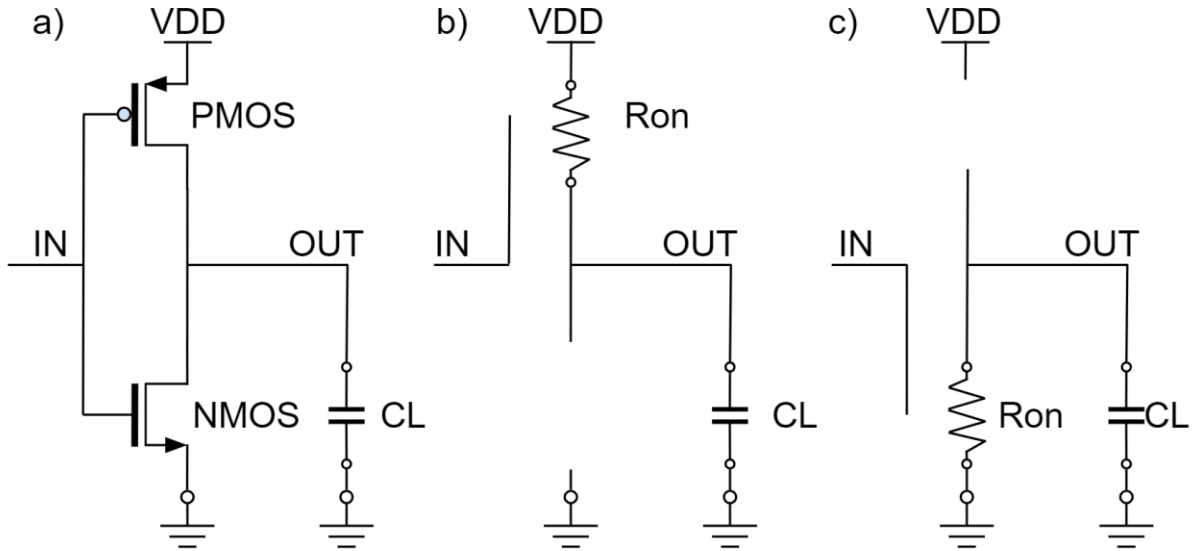
2.2.1 Dynaaminen tehonkulutus

Dynaamisen tehonkulutuksen mallintamiseen voidaan hyödyntää ensimmäisen asteen RC-piiriä tai lineaarista RC-piiriä, sillä lähes kaikki solmupisteet piireissä voidaan ajatella muodostuvan tietyistä kokonaisresistanssista ja kokonaiskapasitanssista. Esimerkiksi kuvan 3 invertterissä PMOS-transistorin ollessa johtavana (kuva 3 b), ladataan lähtösolmun kapasitiivinen kuorma ylemmän resistanssin läpi (avoimen PMOS transistorin kanavaresistanssi R_{on}) ja vastaavasti NMOS-transistorin ollessa johtavassa tilassa (kuva 3 c), kuormakapasitanssin varaus purkautuu maahan vastaavasti NMOS transistorin kanavaresistanssin läpi. Kapasitanssin lataus ja purkaminen aiheuttavat tehonkulutusta piirissä, koska vain niiden aikana virta kulkee R_{on} vastusten läpi. [3] [7]

Ensimmäisen asteen RC-piirin lähtöjännite voidaan laskea kaavalla [3]

$$V_0 = V_{DD} \cdot \left(1 - e^{-\frac{t}{\tau}}\right), \quad (1)$$

Missä V_{DD} on käyttöjännite, V_0 on lähtöjännite, t on kulunut aika ja $\tau = R_{on}C_L$ on piirin aikavakio. Kuvan 3 (a) invertteri kuluttaa energiaa vain sinä aikana, kun kuormakapasitanssia C_L ladataan eli tällöin käyttöjännitteestä otetaan virtaa ja lähtösignaali siirtyy nollassa käytöjännitteeseen. Osa tästä energiasta poltetaan PMOS-transistorissa latausvaiheessa ja osa NMOS-transistorissa purkuvaiheessa. Kondensaattorin varausta purettaessa vain siihen ladattu energia ajetaan maahan, tämä vaihe ei kuluta energiaa käyttöjännitteestä. Yleensä transistorit pyritään mitoittamaan siten, että niiden aktiiviresistanssit ovat yhtä suuret, jolloin lähtösolmussa näkyy molemmissa tilanteissa samankokoinen resistanssi eli piirin aikavakio pysyy samana koko ajan. Aikavakio vaikuttaa piirin signaalien nousu- ja laskuaikoihin sekä viiveeseen, joten oikein mitoitettu rakenne antaa symmetriset vasteet sekä nousevalle, että laskevalle signaalille.



Kuva 3. a) Invertterin piirirakenne PMOS ja NMOS transistoreilla esitettynä. b) Invertterin RC-malli PMOS transistorin ollessa johtavana, kuormakapasitanssin lataus. c) Invertterin RC-malli NMOS transistorin ollessa johtavana, kuormakapasitanssin varauksen purkaminen.

Hetkellinen tehonkulutus RC-piirille voidaan laskea kaavasta

$$p(t) = v(t) \cdot i(t) = V_{DD} \cdot C_L \cdot \frac{dV_0}{dt}, \quad (2)$$

missä $p(t)$ on teho, $v(t)$ on jännite ja $i(t)$ on virta ajanhetkellä t , V_0 on portin lähtöjännite, C_L on lähtösolmun kuormakapasitanssi ja V_{DD} on käyttöjännite. Koska logiikassa jännite on joko 0 V tai V_{DD} , voidaan hetkellinen virta esittää kondensaattorin virtayhtälön avulla. Integroimalla teho ajanjakson t yli, saadaan laskettua käyttöjännitteestä otettu energia, sillä $[W] \cdot [s] = \frac{[J]}{[s]} \cdot [s] = [J]$

$$E = \int_0^t p(t) dt = \int_0^\infty V_{DD} \cdot C_L \cdot \frac{dV_0}{dt} dt = V_{DD} \cdot C_L \int_0^{V_{DD}} dV_0 = C_L V_{DD}^2. \quad (3)$$

Kaavasta (3) nähdään hyvin selkeästi, että suurikokoiset komponentit, joilla on iso kapasitanssi, kuluttavat paljon energiaa. Energian kulutuksen rajoittamiseksi selkein tapa on siirtyä matalampaan käyttöjännitteeseen, sillä sen vaikutus näkyy energiankulutuksessa neliöllisenä. Kun käyttöjännite puolitetaan, energiankulutus pienenee neljäsosaan. Kuormakapasitanssiin varastoitu energia on symmetrisessä rakenteessa puolet käyttöjännitteestä otetusta energiasta – toinen puoli palaa kanavaresistanssissa lämmöksi latauksen aikana.

$$E_{C_L} = \int_0^\infty i(t) V_0 dt = \int_0^\infty C_L \cdot V_0 \cdot \frac{dV_0}{dt} dt = C_L \cdot \int_0^{V_{DD}} V_0 dV_0 = \frac{C_L \cdot V_{DD}^2}{2}, \quad (4)$$

Kertomalla kokonaisenergiankulutus (3) kytkentätaajuudella, saadaan tietyn portin dynaaminen tehonkulutus laskettua, $[J] \cdot [Hz] = [J]/[s] = [W]$. Kytkentätaajuus (=kellotaajuus) kertoo, kuinka monta kertaa sekunnissa piiri ottaa energiaa käyttöjännitteestä. [3]

$$P_{KytKentä} = C_L \cdot V_{DD}^2 \cdot f. \quad (5)$$

Kaavasta (5) nähdään, että nopeat komponentit aiheuttavat myös tehonkulutuksen kasvua, sillä niiden kytkentätaajuus voidaan nostaa hyvin korkeaksi esimerkiksi prosessoreissa.

Toinen tehonkulutukseen vaikuttava osatekijä aiheutuu siitä, kun tilanvaihdon aikana on usein hetki, jolloin molemmat PMOS ja NMOS transistorit johtavat yhtä aikaa. Tällöin piiriin syntyy hetkellinen reitti käyttöjännitteestä maahan eli oikosulku. Virta kulkee invertterin tapauksessa niin kauan kuin tulojännite on korkeampi kuin NMOS transistorin kynnysjännite ja matalampi kuin PMOS transistorin kynnysjännite. Tämä on yleensä hyvin lyhyt aikaväli, sillä kynnysjännitteet pyritään symmetrisen toiminnan vuoksi saamaan mahdollisimman lähelle toisiaan. Oikosulusta aiheutuvaa tehonkulutusta voidaan mallintaa yhtälöllä [8]

$$P_{Oikosulku} = K \cdot (V_{DD} - 2 \cdot V_{th})^3 \cdot \tau \cdot f, \quad (6)$$

missä f on kytkentätaajuus, τ on tulosignaalin nousu- tai laskuaika, V_{th} on kynnysjännite, V_{DD} on käyttöjännite ja K on transistorien koosta ja käytetystä teknologiasta riippuva vakio. Taajuuden kasvaessa lisääntyy myös oikosulkutehonkulutus. Oikosulkutehonkulutus on osa dynaamista tehonkulutusta. Pitämällä tulosignaalin nousuaika samansuuruisena kuin se on lähtösignaalilla, voidaan varmistaa, ettei lähtö kerkeä vaihtaa tilaansa ennen tulosignaalin siirtymistä kokonaan logiikkatasoon. Tällä tavalla oikosulkutehonkulutukseen voidaan vaikuttaa.

2.2.2 Staattinen tehonkulutus

Kolmas tehoon yhä enenevissä määrin vaikuttava seikka on staattinen virrankulutus. Kun piirien koko pienenee, myös käyttöjännitettä muutetaan alemmaksi, jotta tehonkulutus pysyy hallinnassa. Tämä puolestaan aiheuttaa sen, että transistorien kynnysjännitteitä täytyy madaltaa virrananto- ja suorituskyvyn parantamiseksi. Kynnysjännitteen madaltaminen kuitenkin lisää vuotovirtaa heikosta inversiosta johtuen [6]. On olemassa useita erilaisia vuotovirtatyyppejä, esimerkiksi hilaoksidista aiheutuvat erilaiset tunneloitumisilmiöt, käänteisistä pn-liitoksista aiheutuva ja alakynnysjännitteestä (subthreshold) aiheutuva vuotovirta. Staattinen tehonkulutus aiheutuu transistoreiden vuotovirroista ja voidaan esittää kaavalla

$$P_{Vuoto} = I_{Vuoto} \cdot V_{DD}, \quad (7)$$

missä I_{vuoto} on kaikkien vuotovirtamekanismien aiheuttama kokonaisvirta. Kaavasta nähdään, että vuotovirta tapahtuu juuri käyttöjännitteen ja maatason välillä, missä logiikassa ei ideaalitalanteessa ole signaalireittiä. Todellisuudessa transistoreilla on johtamattomanakin joitakin mega-ohmeja suuruusluokaltaan oleva resistanssi. [6]

Mitä alemmas käyttö- ja kynnysjännitteitä lasketaan, sitä isompi osa kaikesta tehonkulutuksesta aiheutuu vuotovirtojen kautta. Virrankulutus lisääntyy myös aktiivimoodissa. Tämän vuoksi vuotovirtoihin pyritään vaikuttamaan joko prosessiteknisesti tai piirisuunnittelun kautta. Valmistusprosessissa vuotovirtaan vaikuttavia asioita ovat esimerkiksi johtavan kanavan mitat, dopppausprofiili ja hilaoksidin paksuus. Usein transistoreista tehdäänkin useiden kynnysjännitteiden malleja, joilla on erilaiset vuotovirrat. Piirisuunnittelussa voidaan vaikuttaa tehonkulutukseen esimerkiksi kytkemällä transistoreja sarjaan tai käyttämällä piireissä useita eri kynnysjännitteen transistoreja, hyödyntämällä transistorien toisen hilan (Bulk) biasointia kynnysjännitteen aktiivisen säätämisen kannalta sekä

säätämällä käyttöjännitettä tilan mukaan. Käyttöjännite voidaan esimerkiksi sammuttaa kokonaan silloin kun piiri siirtyy valmiustilaan. Nykyisten teknologioiden digitaalisuunnittelijat joutuvatkin miettimään piirejä usealla eri tasolla niiden käytöstä riippuen. [6] [9] [10]

2.2.3 Tehonkulutuksen vaikutuksia suunnitteluun

Pienentyvä teknologia on pakottanut suunnittelemaan piirejä useisiin eri käyttötarkoituksiin, jolloin kirjastoista yleensä löytyy pienen vuotovirran omaavia komponentteja ja eri nopeuksille tarkoitettuja komponentteja. Kirjastoista voi myös löytyä komponenteille omia tehomallinnus- ja vuotovirtamallinnustekniikoita. Isoja digitaalipiirejä suunnitellessa usein käytetään aikakriittisissä paikoissa nopeita, enemmän tehoa kuluttavia transistoreja ja muissa paikoissa transistoreja, joiden staattinen tehonkulutus on pientä, eli niiden vuotovirrat ovat matalia.

Tehonkulutus on ajanut piiripinta-aloja pienemmiksi ja käyttöjännitteitä alemmaksi. Pinta-alan pienentäminen tarkoittaa myös useimmiten sitä, että samalla pinta-alalla on enemmän komponentteja, jolloin tehonkulutus pinta-ala yksikköä kohden kasvaa. Tämä pakottaa myös miettimään jäähdytysmekanismeja piireille uudella tavalla. Tässä työssä ei tarkastella suunnittelua kovinkaan korkealla tasolla, sillä tarkoitus on suunnitella vain yksittäisiä, muutamista transistoreista koostuvia piirejä. Jäähdyttäminen ja muut siihen liittyvät asiat tulevat kyseeseen silloin, kun suunnitellaan satoja tuhansia tai miljoonia transistoreja sisältäviä IC-piirejä. Suuriin ja tehoa vaativiin IC-piireihin suunnitellaan myös usein sisäisiä suojausmekanismeja, kuten virtarajoittimia, yli- ja alijännitesuojia sekä piirejä, joilla voidaan estää ylikuumeneminen [11].

2.3 Kohinamarginaalit

Häiriöitä on aina. Niitä tulee esimerkiksi induktiivisen ja kapasitiivisen kytkeytymisen kautta ja käyttöjännitteiden mukana. Logiikka on siitä hienoa, että se omalta osaltaan puhdistaa signaaleja. Puhdistuminen perustuu siihen, että pienet jännitevaihtelut eivät aiheuta logiikkatason vaihtumista, vaan huolimatta tulon häiriöstä, lähdössä näkyy siisti signaali. Kuvassa 4 on esitettyä kaavio, josta voidaan määrittää kohinan rajat signaalissa. Kohinaraja tarkoittaa sitä kohinan jännitteen rajaa, jolla piirin normaali toiminta ei häiriinny. Kaavion perusteella ajavan laitteen lähdön logiikkatason 1 minimijännite tulee olla ajettavan laitteen tulosignaalin logiikkatason 1 minimijännitteen yläpuolella. Tästä saadaan ylemmäksi kohinarajaksi

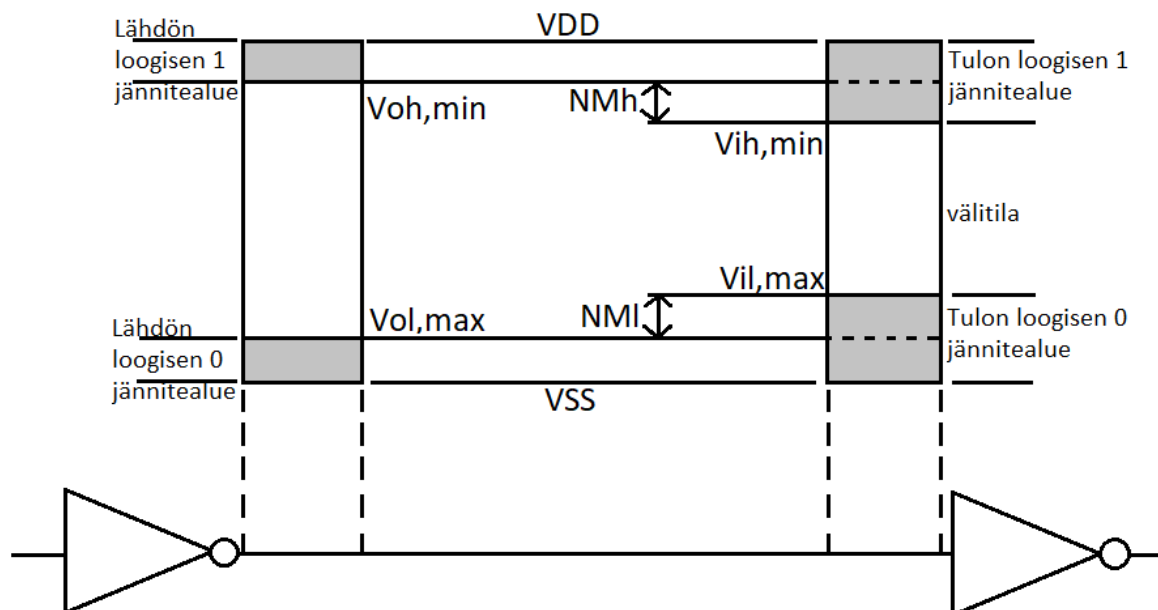
$$NM_H = |V_{OHmin} - V_{IHmin}|, \quad (8)$$

Vastaavasti voidaan määrittää myös alempi kohinaraja, jossa ajavan laitteen lähdön logiikkatason 0 maksimijännite tulee olla ajettavan laitteen tulon logiikkatason 0 maksimijännitteen alapuolella.

$$NM_L = |V_{OLmax} - V_{ILmax}|, \quad (9)$$

Kaavoissa (8) ja (9) NM on kohinamarginaali, V_{OHmin} on lähdön korkeamman potentiaalin minimi, V_{OLmax} on lähdön matalamman potentiaalin maksimi, V_{IHmin} on tulon korkeamman potentiaalin minimi ja V_{ILmax} on tulon matalamman potentiaalin maksimi. Käytännössä kohinaa

kestää olla signaaleissa sen verran, ettei kohina aiheuta logiikkatason muuttumista. Kohinamarginaaleja voidaan kasvattaa käyttämällä schmitt-triggeriä. [3]



Kuva 4. Kohinarajojen määrittäminen. Kaavion alareunassa olevat invertterit havainnollistavat kohinarajojen paikat logiikkaporttien suhteeseen.

2.4 Komponenttikirjastot

Digitaalilogiikkakirjasto on tietyllä logiikkaperheelle rakennettu kokoelma komponentteja, joiden avulla suunnittelija voi rakentaa omia piirejä. Kirjaston komponenteista löytyy valmiit piirikaavio- ja piirikuviomallit, joita suunnittelija hyödyntää oman digitaalisen IC-piirin rakentamiseen. Komponenttikirjastosta löytyy logiikkakomponenttien perusportit ja joitakin monimutkaisempia rakenteita sen verran, että kirjaston käyttäjä voi niiden avulla rakentaa melkein mitä tahansa toiminnallisuuksia.

Digitaalilogiikkakirjastosta voi löytyä erilaisiin vaatimuksiin soveltuvat komponentit, kuten mahdollisimman nopeat tai hyvin pienen vuotovirran omaavia logiikkaportteja, sekä mahdollisesti näiden välistä. Nämä kirjaston eri osat sisältävät yleensä samat logiikkaportit, jotka on mitoitettu hieman eri tavalla. Komponenttikirjastoissa on yleensä käyty läpi eri nopeusasteen ja vuotovirran komponenttimallit prosessinvaihtelun ja lämpötilojen vaikutuksien kanssa. Jokaiselle kirjaston osa-alueelle on suunnittelija tehnyt dokumentit, joista selviää, mitä kynnysjännitteitä, käyttöjännitteitä ja muita ominaisuuksia mikäkin transistorimalli vastaa. Hyvin usein kirjastoista löytyy mallit LVT (Low V_T), RVT (Regular V_T) ja HVT (High V_T). [10] [12]

2.4.1 Prosessi-, jännite- ja lämpötilavaihtelut

Saman transistorin ominaisuudet voivat vaihdella valmistusprosessin tuomien muuttujien ansiosta. Samalla piikiekolla olevien transistorien kynnysjännitteet voivat olla eri suuruiset tai peräkkäisinä päivinä valmistettujen piirien ominaisuudet voivat poiketa toisistaan, sillä valmistusprosessin aikana ei voida olla varmoja, että jokainen transistori saa yhtä paksut oksidikerrokset, metallikerrokset ja muut ominaisuudet. Prosessivaihtelu voi olla satunnaisesti tapahtuvaa, liittyä sijaintiin piikiekolla tai tiettyihin rakenteisiin, vaihtelua voi olla saman piirin

sisällä eri elementtien välillä tai samalle piikiekolle valmistetut piirit voivat olla erilaisia [13]. Usein vaihteluista puhutaan PVT-vaihteluina, eli prosessi-, jännite- ja lämpötilavaihteluna, joista jokainen aiheuttaa toimintaan muutoksia. Näitä toiminnan muutoksia pyritään usein kompensoimaan eri menetelmillä [10] [14]. Lähteessä [10] on käsitelty rungon biasoinnin vaikutuksia esimerkiksi vuotovirtaan, toimintataajuuteen ja tehonkulutukseen sekä PVT vaihteluihin.

Komponentteja suunnitellessa pyritään usein ottamaan huomioon näitä muuttujia. Komponenttimalleille voi olla dokumentoituna simulointitulokset, joissa näkyy esimerkiksi nurkkasimulointien, kuten FF (Fast-Fast), TT (Typical-Typical) ja SS (Slow-Slow), vaikutukset. Nämä kolme nurkkaa ovat hyvin yleisesti käytössä, mutta myös muitakin, kuten FS (Fast-Slow) ja SF (Slow-Fast), voidaan käyttää esimerkiksi silloin kun käytetään resistiivisiä ylös vetoja. Usein komponentit simuloidaan eri prosessinurkissa, lämpötiloissa ja erilaisilla käyttöjännitteillä, jotta saadaan kattava kuva toiminnasta. Näin käyttäjälle on valmiina esitetty, miten mikään komponentti toimii missäkin olosuhteissa ja valinta on helpompi tehdä. [15] [16]

Digitaalisia IC-piirejä suunnitellessa voidaan noudattaa ns. worst case suunnittelumenetelmää, eli suunnitellaan piirit siten, että ne toimivat annettujen rajojen ääriladoissa, joten ne todennäköisesti toimivat myös rajojen välissä [17, s.237-249]. Nanometriluokan CMOS-teknologioissa kuitenkin pahimmat tapaukset ovat hankalia ennustaa pelkästään prosessinurkkien tai lämpötilojen avulla, koska komponentit ovat tässä kokoluokassa hyvin herkkiä melkein mille tahansa vaihtelulle. Piireistä tehdäänkin pienissä teknologioissa hyvin usein adaptiivisia, eli ne pyrkivät mukautumaan vaihteluihin ja toimimaan optimaalisella alueella koko ajan. [18]

2.4.2 Vuotovirta- ja kynnysjännite

Transistorien vuotovirtoihin ja kynnysjännitteisiin voidaan vaikuttaa valmistusprosessissa fyysisten ominaisuuksien kautta. Esimerkiksi kynnysjännitteeseen vaikuttavia asioita ovat valmistuksessa käytetty hilamateriaali, transistorin hilan alla olevan oksidikerroksen paksuus, substraatin douppauskonsentraatiot, varauskonsentraatiot, kanavan pituus ja leveys sekä mahdollinen rungon biasjännite. Erilaisten vuotovirta- ja kynnysjännitemallien suunnittelu onkin tapahtunut muokkaamalla transistorin fyysisiä ominaisuuksia ja ne ovat piirivalmistajan toteuttamia. [19]

Logiikkaa suunnitellessa on hyvä ajatella asioita tehonkulutuksen ja nopeuden näkökulmasta. SLVT (Super-Low V_T) ja LVT ovat matalan kynnysjännitteen malleja, jotka mahdollistavat nopeiden piirien suunnittelun. Komponentit ovat hyvin pieniä, jolloin niillä on myös pienet kapasitanssit porteissa. Toisaalta näissä komponenteissa on myös yleensä isompi vuotovirta, mikä aiheuttaa staattiseen tehonkulutukseen kasvua [10]. HVT ja UHVT (Ultra High V_T) ovat korkean kynnysjännitteen malleja, joissa vuotovirta on pyritty puristamaan minimiin. Näitä komponentteja on digitaalisuunnittelussa hyvä käyttää paikoissa, joissa aika, nopeus ja pinta-ala eivät ole ratkaisevia. RVT on eräänlainen välimalli, joka kuvaa normaalia toimintaa.

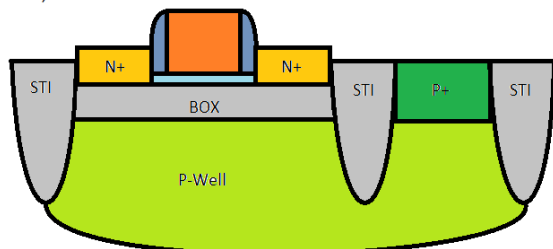
Erilaisten kynnysjännitemallien perusteella voidaan päätellä, että käyttämällä eri kynnysjännitteen malleja saadaan esimerkiksi tehonkulutusta jaettua sijoittamalla matalan ja korkean kynnysjännitteen transistorit omiin lohkoihinsa. Vastaavalla tavalla myös ajoituskriittisissä lohkoissa voidaan hyödyntää nopeita transistoreja ja muualla vähän hitaampia.

2.5 FD-SOI Prosessi

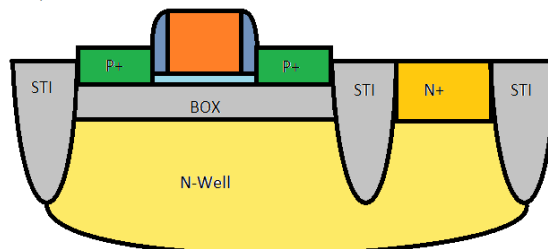
FD-SOI (Fully Depleted Silicon On Insulator) on prosessi, jossa hyvin ohut johtava kanava on eristetty substraatista/altaasta täysin käyttämällä hyväksi haudattua oksidikerrosta (Buried Oxide, BOX). Tämä rakenne vähentää nielun ja lähteen parasiittisia ominaisuuksia, sekä pienentää vuotovirtaa. Yksi rakenteen tuomia etuja on myös altaan mahdollinen biasointi ja sitä kautta kynnysjännitteen siirtäminen haluttuun suuntaan. Biasointi on mahdollista, koska rakenteesta puuttuu suorat nielu- ja lähde-substraatti-liitokset. Substraatti toimii käytännössä toisena hilana ja toimii samaan suuntaan kuin alkuperäinenkin. Altaan tyyppi tulee valita niin, ettei sitä biasoida samaan suuntaan oikean hilan kanssa. Toisen hilan biasointialue riippuu käytetyn rakenteen p- ja n-altaiden välille muodostuvan diodin kynnysjännitteestä sekä läpilyöntijännitteestä ja voi olla suuruusluokkaa $0\text{ V} \pm 3\text{ V}$. [20] [21] [22]

Prosessissa käytetty BOX kerros, jolla eristetään johdinkanava altaasta, mahdollistaa sen, että sekä PMOS rakenteet, että NMOS rakenteet voidaan tehdä n- ja p-tyypin altaiden päälle. Kun tehdään normaali tyylillä komponentteja, rakennetaan NMOS p-altaan päälle ja PMOS n-altaan päälle, jolloin suunnitellaan RVT-mallin komponentteja. Tällöin altaan vastasuuntaisella biasoinnilla (RBB- Reverse Body Bias), jossa NMOS altaan jännitettä lasketaan ja PMOS altaan jännitettä nostetaan, voidaan vähentää vuotovirtaa hyvin pieneen osaan alkuperäisestä. Vastaavasti, kun käytetään käänteistä rakennetta, suunnitellaan LVT-mallin komponentteja. Näissä biasoimalla allas samaan suuntaan kuin hilajännitekin, voidaan kasvattaa komponentin nopeutta hyvinkin paljon. Käänteisellä rakenteella pienin tehonkulutus saavutetaan, kun $|V_{BG}| = 0$. Tällöin kynnysjännite pysyy kohtuullisen korkeana ja vuotovirta on pieni. Jos $|V_{BG}| > 0$, saadaan korkeampi saturaatiovirta ja komponentin resistanssi pienenee (R_{ON}), mikä johtaa pienempään kynnysjännitteeseen ja tätä kautta lisää nopeutta. Kuva 5 havainnollistaa FD-SOI prosessin mahdollisia MOS-rakenteita. [20] [21] [22] [23]

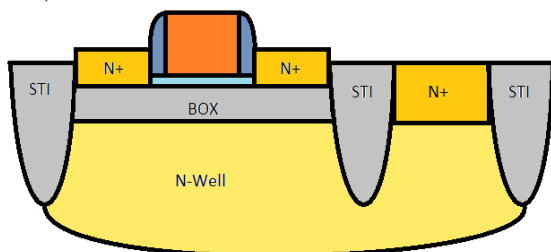
a) NMOS, normaali rakenne



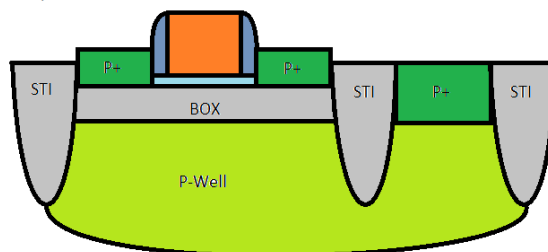
c) PMOS, normaali rakenne



b) NMOS, käänteinen rakenne



d) PMOS, käänteinen rakenne



Kuva 5. FD-SOI rakenteen poikkileikkaus [20] [22], jossa NMOS tavanomaisella altaalla (a) ja käänteisellä altaalla (b) sekä PMOS tavanomaisella altaalla (c) ja käänteisellä altaalla (d).

3 PIIRIKAAVIOIDEN SUUNNITTELU

Kirjastokomponenttien piirikaavioiden suunnittelussa hyödynnettiin yleisesti tunnettuja rakenteita. Varsinaista toimintalogiikkaa ei tarvinnut kehittää uudestaan, mutta mitoitus täytyi tehdä jokaiselle komponentille tarpeen mukaan. Komponenttien suunnittelussa ja transistorien mitoituksessa iso rooli on käytössä olevalla teknologialla ja valmistusprosessilla. Ne määräävät käytännössä millaisia jännitteitä ja virtoja mihinkin toiminnalliseen lohkoon pystytään ajamaan. Suunnittelussa on lisäksi otettu huomioon 22 nm FD-SOI-prosessin tuomat virta- ja jänniterajoitukset eri metallikerroksille. Mitoitusprosessi on kaikille komponenteille ollut suurin piirtein samanlainen, eli pyritään saamaan nousu- ja laskuajat mahdollisimman symmetrisiksi sekä viiveet mahdollisimman pieniksi. Komponentit eivät myöskään saa aiheuttaa liikaa häiriötä tai logiikkatasojen muuttumista muualla piirillä tapahtuvien transienttien vaikutuksesta.

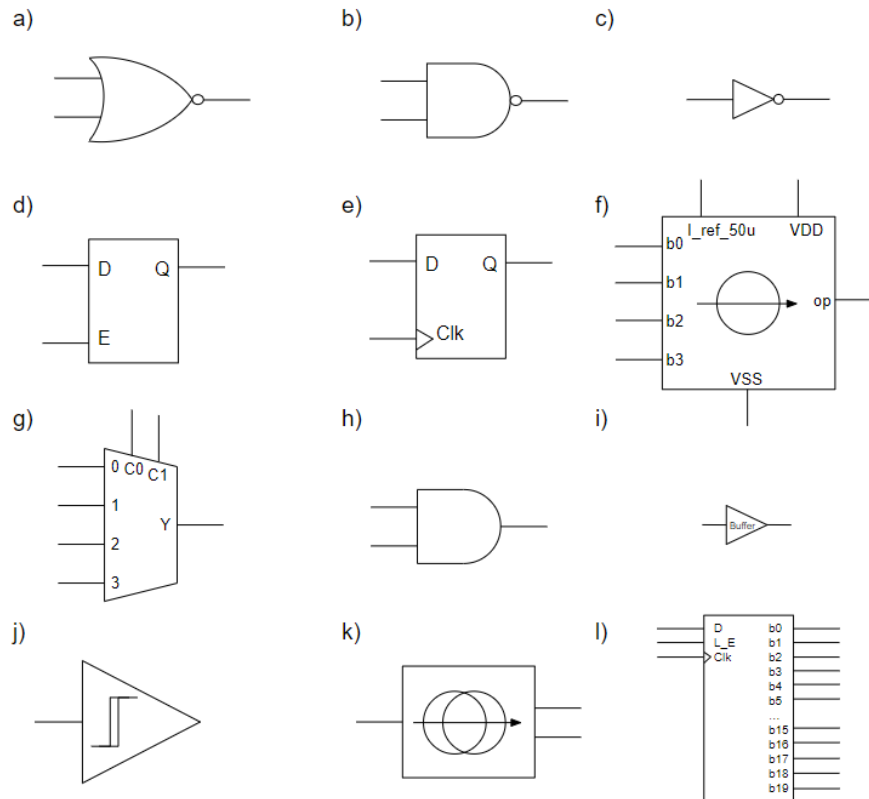
3.1 Työn tavoitteet ja tarvittavat komponentit

Logiikan suunnittelussa pyritään luomaan piirejä, joiden tehonkulutus ja pinta-ala ovat pieniä, sekä toiminta riittävän nopeaa. Nämä asiat ovat myös tämän työn tavoitteena ja siksi suunnittelussa käytettiin pieniä ja nopeita SLVT-tyypin transistorimalleja. Työssä ei ollut tarvetta rakentaa useiden nopeus- ja vuotovirtamallien kattavaa kirjastoa, joten kokoelma rakennettiin vain yhdelle kynnysjännitemallille. Tarkemmat spesifikaatiot, lämpötilan ja käyttöjännitteen vaihtelut sekä prosessinurkat jätettiin vähemmälle huomiolle, koska rakenteet ovat yksinkertaisia ja ei-kaupallisessa tarkoituksessa tehtyjä. Ohjauksen kannalta tärkeille komponenteille, kuten kiikulle, rekistereille ja virta-DAC:lle tehtiin lämpötila- ja prosessinurkkasimuloinnit. Simuloinneissa saatiin riittävän kattava kuva, kun simuloitiin nurkat FF lämpötilassa -15°C, TT lämpötilassa 27°C ja SS lämpötilassa 85°C. Myös prosessissa mahdollinen alaan biasointi jätettiin pois, sillä se olisi vaatinut hieman enemmän pinta-alaa haudattujen kerrosten vuoksi ja rakenteet olisivat monimutkaistuneet.

Tässä työssä rakennettu kirjasto sisältää tarvittavat komponentit piirin analogia- ja RF-komponenttien biasten ohjaukseen sekä mahdollistaa analogisten kytkimien käytön digitaalisen ohjauksen avulla. Se sisältää logiikan peruskomponentteja, kuten NOT-, NOR- ja NAND-portteja ja kiikkuja, mutta myös vähän mutkikkaampiakin komponentteja, kuten peruskomponenttien avulla toteutettuja AND-portteja, siirtorekistereitä, multipleksereitä ja virta-DAC (Current Digital to Analog Converter).

Suunnittelussa hyödynnettiin hierarkia-ajattelua. Ensin peruskomponenteista luotiin transistorirakenteet, jotka mitoitettiin halutulla tavalla. Mitoituksen jälkeen peruskomponenteille luotiin symbolit, joita hyödynnettiin suoraan ylemmän tason piirikaavioissa. Monimutkaisemmille piireille luotiin myös omat symbolit, mutta piirien mitoituksen ja pinta-alan optimoimiseksi osa monimutkaisemmista piireistä rakennettiin kuitenkin kustomoituna transistoritasolla. Kun kullekin komponentille on oma symboli, voi suunnittelija hyödyntää paljon helpommin kyseisiä komponentteja.

Kuvassa 6 nähdään tässä työssä rakennettujen komponenttien symbolit. Kuvissa a) b) ja c) on esitetty NOR-, NAND- ja NOT-portin symbolit. Kuvissa d) – l) ovat D-salvan, D-kiikun, virta-DAC:in, multipleksaajan, AND-portin, puskurin, schmitt-triggerin, virtapeilin ja 20-bittisen rekisterin symbolit. Näitä symboleita käytetään tässä työssä esitettyjen piirirakenteiden ja testipenkien osana. Kuvassa esitetyt symbolit ovat havainnollistavia, eikä kuvassa ole esitetty kuin yksi symboli kutakin komponenttia kohden, vaikka kirjasto voi sisältää useita eri versioita samasta komponentista.



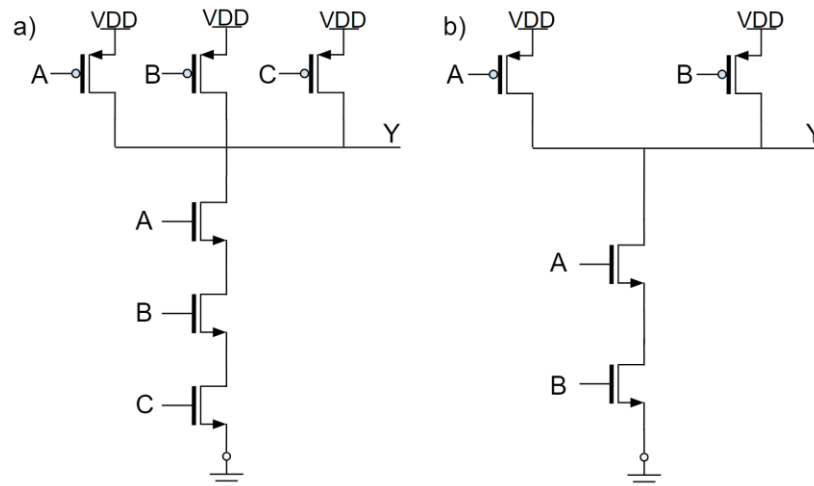
Kuva 6. Eri komponenttien symboleita. a) NOR-portti, b) NAND-portti, c) invertteri, d) D-salpa, e) D-kiikku, f) virta-DAC, g) MUX, h) AND-portti, i) Puskuri, j) Schmitt-triggeri, k) virtapeili ja l) 20-bittinen rekisteri.

3.2 Peruskomponentit ja niiden transistorimallit

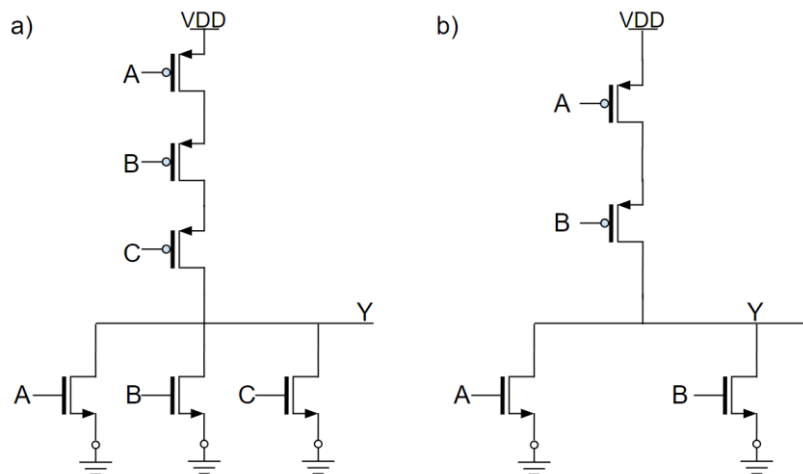
Peruskomponenteiksi luokitellaan ne komponentit, jotka rakennetaan lähes aina logiikkaperheeseen. Näistä komponenteista esitetään transistoritason mallit ja niiden avulla voidaan toteuttaa mitä tahansa muuta logiikkaa.

3.2.1 NAND- ja NOR-portti

Kuvassa 7 ja 8 on esitetty logiikkakirjastojen tärkeimpien perusporttien, NAND- ja NOR-porttien, piirikaaviot sekä 3-tuloisena (a), että 2-tuloisena (b). Peruskomponenttien nimeäminen on suorassa suhteessa tuloporttien määrään, esimerkiksi kuvissa näkyvät portit ovat NAND_X3, NAND_X2, NOR_X3 ja NOR_X2. Perusporttien mitoituksessa lähdettiin liikkeelle siitä, että nousu- ja laskuajat tulisi olla yhtä suuret. Useimmissa teknologioissa PMOS transistorien varausten liikkuvuus on pienempi, kuin NMOS transistoreilla, joten mitoituksessa p-tyypin transistorit voivat olla esimerkiksi 2-kertaa suurempia [2]. Tässä työssä lähes symmetriset nousu- ja laskuajat saatiin kuitenkin aikaiseksi käyttämällä samankokoisia rakenteita.



Kuva 7. a) 3-tuloisen NAND-portin ja b) 2-tuloisen NAND-portin piirikaaviot.



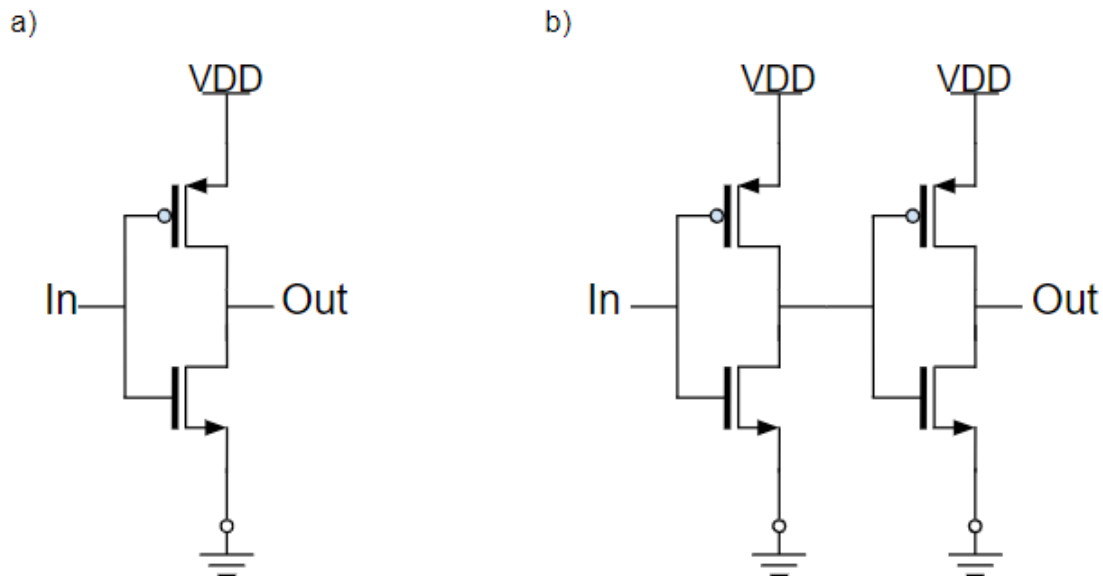
Kuva 8. a) 3-tuloisen NOR-portin ja b) 2-tuloisen NOR-portin piirikaaviot.

3.2.2 Invertteri ja puskuri

Invertterin eli NOT-portin rakenne on yksinkertainen. Siinä on vain yksi NMOS-PMOS pari, joiden hilat on kytketty yhteen ja nielut on kytketty yhteen. Kuvassa 9 a) on esitetty invertterin perusrakenne. Invertterin kokoa voidaan suurentaa kasvattamalla NMOS- ja PMOS-transistorien pituudet ja leveydet isommiksi, lisäämällä sormien lukumäärää tai lisäämällä minimikokoisia transistoreja sarjaan tai rinnan sekä kytkemällä kaikkien hilat yhteen. Transistorien sarjaan kytkeminen hidastaa rakennetta huomattavasti, joten invertterit mitoitettiin lisäämällä sormien lukumäärää. Nimeäminen tapahtuu siten, että invertterin nimen, esimerkiksi INV_X3, loppuosa kertoo koon olevan 3-kertainen suhteessa minimikokoiseen invertteriin. Tarkempi analyysi invertteristä ja sen ominaisuuksista löytyy lähteestä [3, s.180-233].

Kuvassa 9 b) on puskurirakenne, jonka tulopuolella on peruskokoinen NMOS ja PMOS transistorista rakennettu invertteri ja lähtöpuolella toinen vastaavanlainen, mutta suurempi invertteri. Puskureiden nimeämiskäytännössä esimerkiksi BUF_X9 tarkoittaa lähtöpuolen invertterin suhdetta minimikokoiseen invertteriin, tässä tapauksessa 9-kertainen. Puskureissa kokoa kasvatetaan lisäämällä transistorin sormien lukumäärää. Kaikki puskurirakenteet on tehty siten, että tulopuolen ja lähtöpuolen inverttereiden kokosuhte on noin 1:3. Invertterin

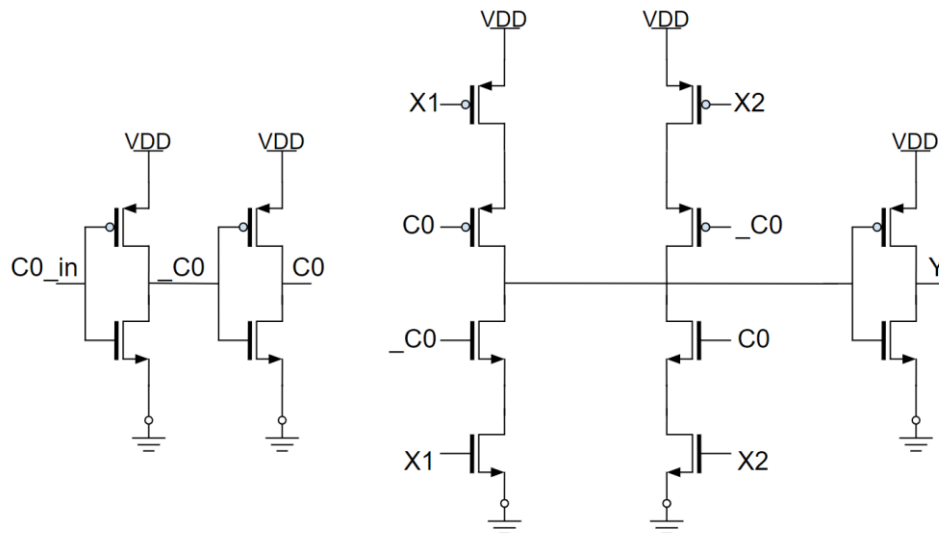
koon kasvattaminen ketjun lähtöä kohti lisää nopeutta ja vähentää kapasitiivisen kuorman vaikutusta puskuroitavaan signaaliin eli mahdollistaa suuremman kapasitiivisen kuorman ajamista [3] [24]. Puskureiden tehtävä on lisätä virranantokykyä vahvasti kapasitiivisiin kuormiin. Niitä voidaan käyttää komponentin tulo- ja lähtösignaalien puskuroimiseen, jolloin esimerkiksi kapasitiivinen kuorma ei vaikuta niin paljon komponentin varsinaiseen toimintaan.



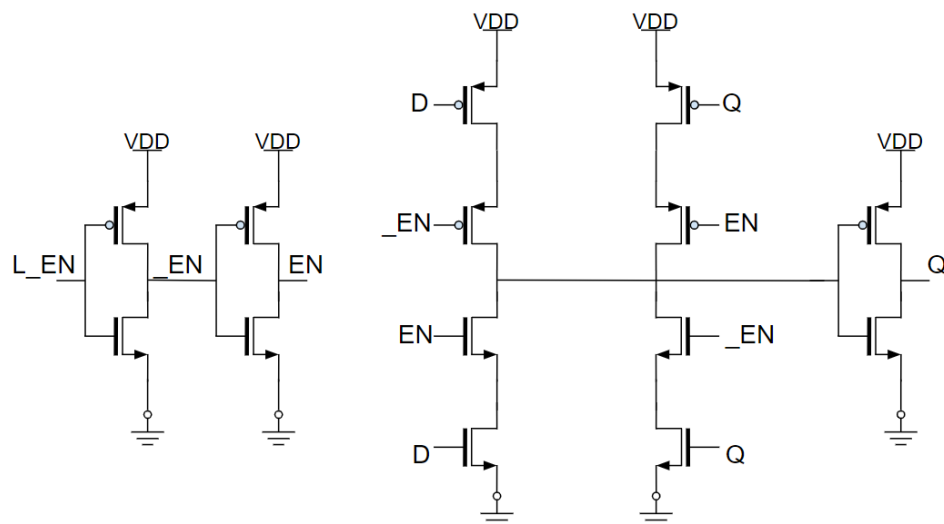
Kuva 9. a) Invertterin ja b) puskurin piirikaaviot.

3.2.3 Multiplekseri, D-salpa ja Schmitt-triggeri

Multiplekseri on komponentti, jolla valitaan yksi tulosaika lähtöön sallintasiinaalin avulla. Kuvassa 10 on esitetty 2-tuloisen MUX_2:1 rakenne. Toiminta perustuu 3-tila inverttereihin, joilla voidaan siirtää tulosaika käännettynä eteenpäin. Tämä puolestaan ajetaan perinteisen invertterin läpi, jolloin lähdössä näkyy valittu tulosaika. Sallintasiinaali on kytketty 3-tila inverttereihin siten, että sen ollessa alhaalla, X1 näkyy lähdössä ja sen ollessa ylhäällä, X2 näkyy lähdössä. Kuvassa 11 on esitetty puolestaan D-salvan rakenne, joka on käytännössä positiivisesti takaisinkytketty 2-tuloisen multipleksaaja. Rakenteessa sallintasiinaali on nyt kytketty siten, että sen ollessa ylhäällä tulosaika näkyy lähdössä ja sen ollessa alhaalla salpa säilyttää tilansa. Toisen 3-tila invertterin tuloon on takaisinkytketty lähtösiinaali Q. Myös puskuri on mahdollista toteuttaa 3-tila invertterin avulla kytkemällä ohjaukset käänteisesti. [3][25]



Kuva 10. 2-tuloisen multiplekserin piirikaavio.

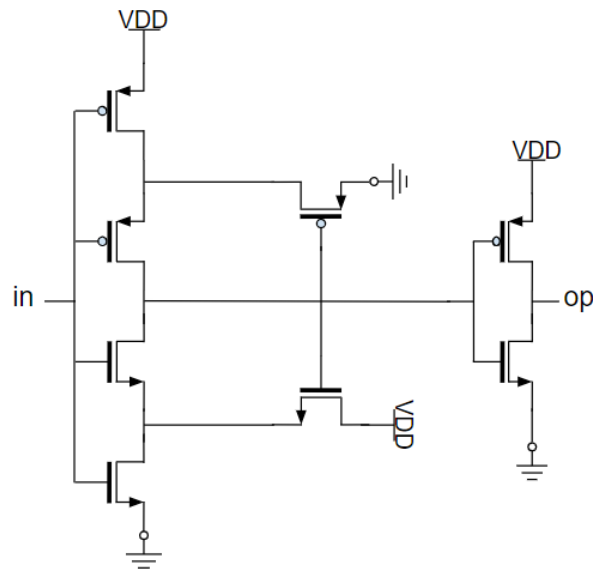


Kuva 11. D-Salvan piirikaavio, jonka rakentamiseen käytetty positiivisesti takaisinkytkettyä multiplekseriä.

Schmitt-triggerillä on kaksi tärkeää ominaisuutta. Se reagoi hitaisiin signaaleihin siten, että lähtö nousee tai laskee jyrkästi tulosignaalin siirtyessä kynnsjännitteen toiselle puolelle ja lisäksi se siirtää kynnsjännitettä ylös ja alas sen mukaan kumpaan suuntaan edellinen siirtymä tulossignaaliissa on ollut. Kuvan 12 rakenne esittää ei-kääntävän schmitt-triggerin piirirakennetta. 3-tila invertterin kaikki hilat on kytketty tulossignaaliin ja lähdöstä on PMOS ja NMOS transistorien kautta takaisinkytkennät inverttereiden n- ja p-tyypin transistorien väliin. Tällöin lähtösignaalin muuttaessa tilaansa ohjaavat nämä kaksi takaisinkytkentää transistorien väliset jännitteet siihen suuntaan, että transistorien kytkeytyminen vaatii erikokoisen kynnsjännitteen siirtymään nähden. 3-tila invertterin lähtö on vielä käännetty, jotta toiminnasta saadaan ei- kääntävä.

Schmitt-triggeriä käytetään tilanteissa, joissa halutaan välttyä tilan vaihtumiselta esimerkiksi kohinan vuoksi tai halutaan loivista nousu- ja laskureunoista muodostaa terävämpiä. Kohinan sietokyky perustuu hystereesiin, eli kun tulojännite ylittää tietyn kynnyksen, lähtö vaihtaa

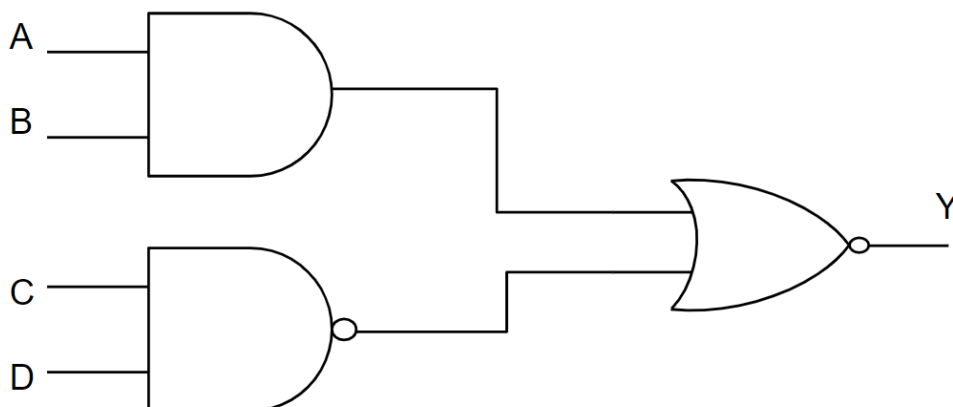
tilaansa ja samalla kynnysjännite muuttuu siten, että tulojännitteen pieni heilahdus takaisin päin ei ylitä/alita uutta kynnystä, jolloin kohinaa ja pieniä häiriöitä saadaan tehokkaasti pois. [3]



Kuva 12. Schmitt-triggerin piirikaavio.

3.3 Peruskomponenttien hyödyntäminen monimutkaisemmissa rakenteissa

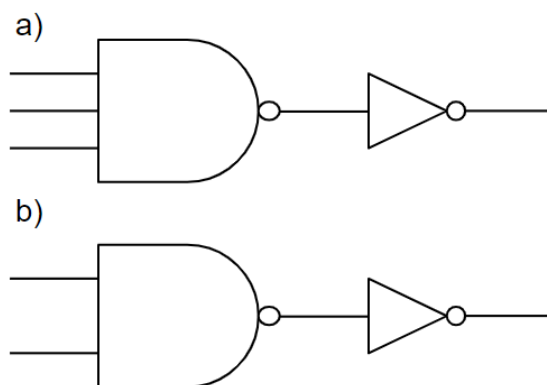
Kirjastoon rakennetut perusportit ovat omiaan suunnittelijalle, joka haluaa luoda omia loogisia operaatioita suorittavia komponentteja. Jokaisen peruskomponentin toiminnallisuuksia yhdistelemällä voidaan rakentaa uutta toiminnallisuutta. Yksi tapa määritellä uusia loogisia operaatioita on Boolean algebra, jossa on logiikan eri operaatioille omat merkinnät ja niistä muodostettujen yhtälöiden avulla rakennetaan piirejä. Logiikan peruskomponenteilla voidaan toteuttaa Boolean algebran lausekkeita suoraan toiminnallisiksi yksiköiksi, kuten esimerkiksi kuvassa 13 esitetty rakenne, joka toteuttaa lausekkeen $Y = \overline{(AB + \overline{CD})}$. [25, s.34-35]



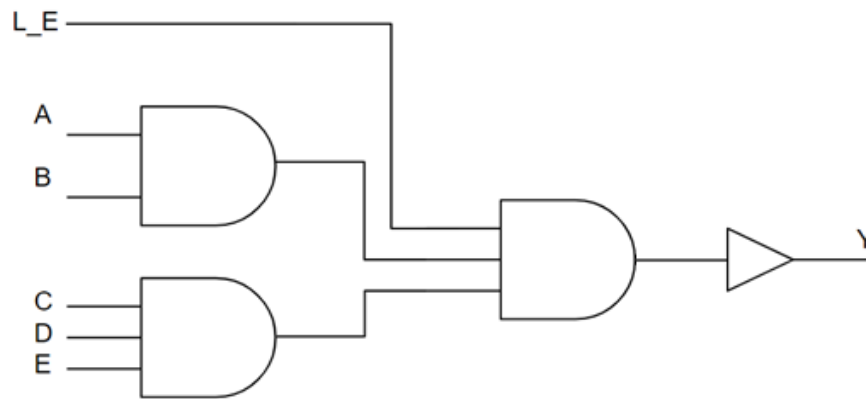
Kuva 13. Esimerkki peruskomponenttien käyttämisestä logiikan rakentamisessa.

3.3.1 AND-portti, siirtorekisteri ja osoitodekooderi

AND-portti toimii siten, että kaikkien tulossignaalien ollessa yhtä aikaa ylhäällä lähtö siirtyy ylös. Sen toiminta on NAND-portille käänteinen, kuten voidaan nähdä kuvasta 14 a) ja b), jossa AND-portti on rakennettu NAND-portista kääntämällä sen lähtösignaali. Kuvan 15 rakenne kuvaa 5-tuloista AND-porttia, jossa on sallintasignaali. 5-tuloisen AND-portin tekemiseen hyödynnettiin kuvan 14 AND-portteja sekä puskuria. [2]

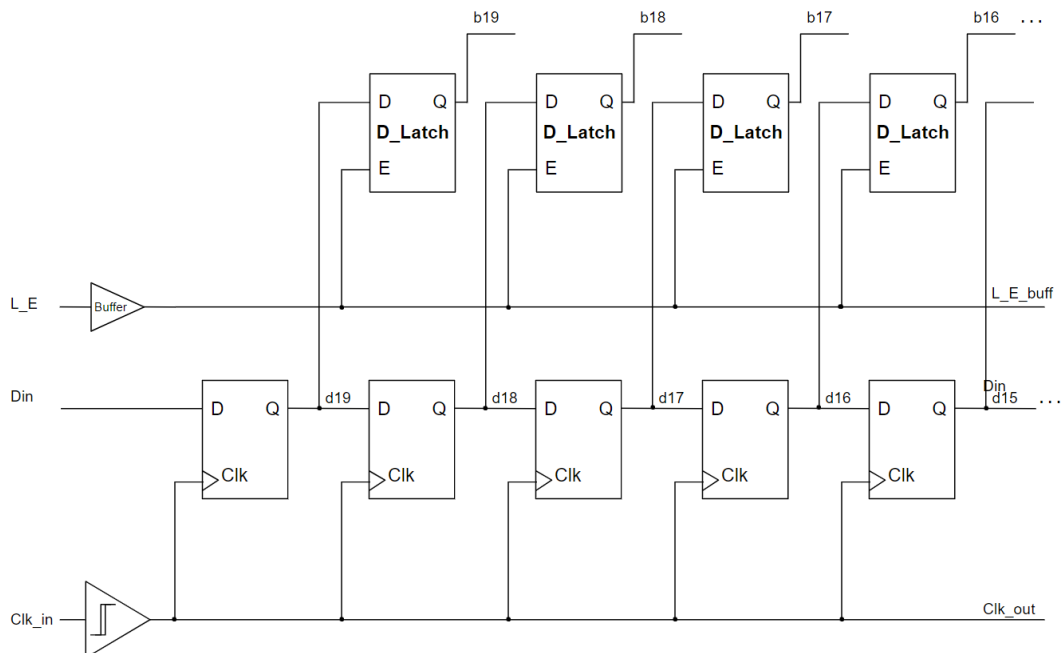


Kuva 14. Lohkokaaviot a) 3-tuloinen AND-portti b) 2-tuloinen AND-portti.



Kuva 15. 5-tuloisen AND-portin lohkoakaavio salpasallinta signaalilla.

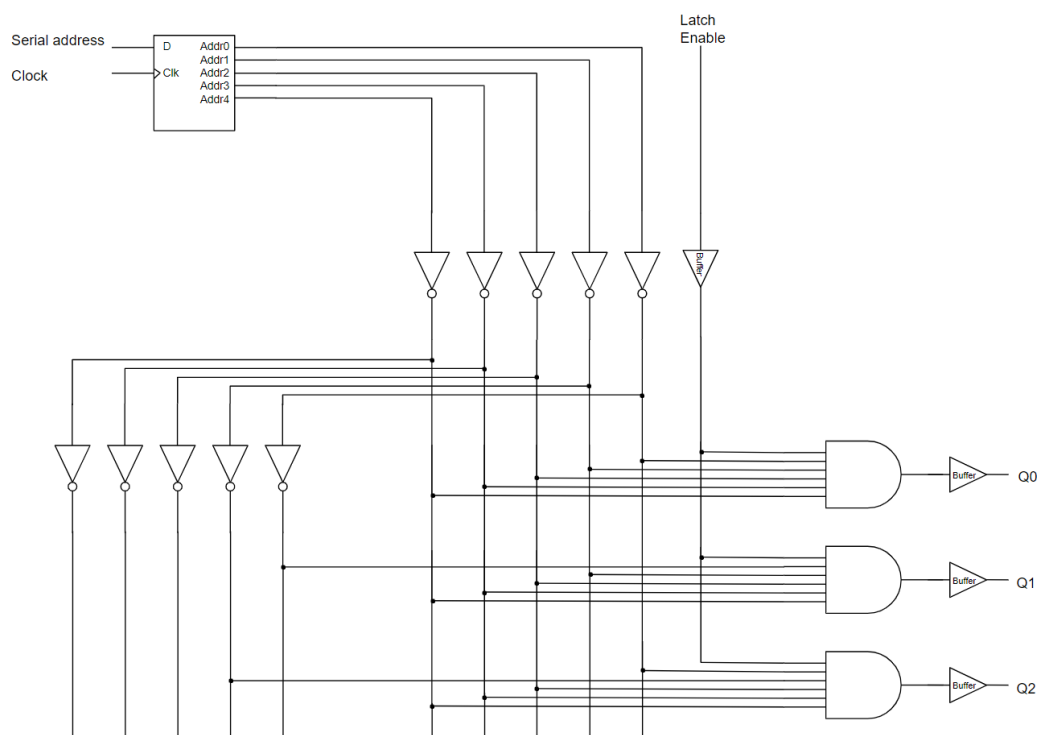
Peruskomponenteista voi myös rakentaa monimutkaisempiakin rakenteita, kuten vaikkapa kuvassa 16 esitetty siirtorekisteri, jota voidaan käyttää sarjamuotoisen datan muuttamiseksi rinnakkaiseen muotoon. Salpojen ohjaussignaali voi halutulla ajanhetkellä ladata siirtorekisterin kiikkujen bitit lähtöön rinnakkaisina. Siirtorekisteri voi myös päästää datan sarjamuotoisena läpi yhdestä lähtöportista. Usein tällaisissa rakenteissa on kontrollisignaali, jolla voidaan valita, meneekö data sarjamuotoisena rekisterin läpi vai ei. Joillakin siirtorekisterirakenteilla voi myös muuttaa rinnakkaisen datan sarjamuotoiseksi dataksi. Rekistereissä kellosignaalin täytyy pysyä terävänä, joten se ajetaan schmitt-triggerin läpi. [17] [25] [26]



Kuva 16. Osa siirtorekisterin lohkoakaaviosta.

Toinen hieman monimutkaisempaa logiikkaa vaativa komponentti on osoitedekooderi, jonka rakennetta on havainnollistettu kuvassa 17. Siinä on tulopuolella 5-bittinen siirtorekisteri,

joka muuttaa sarjamuotoisen osoitteen rinnakkaiseksi, ja kombinaatiologiikkaa, joka mahdollistaa osoitteen mukaan valikoituvan lähtösignaalibitin Q0-Q31. Kuvassa ei näy lähtöporteista kuin 3 ensimmäistä. Kun tuloon tuodaan tietty osoite, dekooderi nostaa kyseistä osoitetta vastaavan lähtöbitin ylös ja haluttuun rekisteriin voidaan ladata uutta dataa. Lähtösignaali määräytyy suoraan tulobittien arvosta, esimerkiksi bittisana 10011 nostaa lähdön Q19 ylös, jos sallintasignaali on päällä. Osoitedekooderit nimetään usein tulo-lähtöbittien suhteessa $K\text{-to-}2^K$, missä K on rinnakkaisten tulobittien määrä. Tässä työssä rakennettiin sarjamuotoinen $5\text{-to-}2^5$ eli $5\text{-to-}32$ osoitedekooderi. [3] [27]



Kuva 17. Osa 5-32 osoitedekooderin lohkokaaviosta.

3.3.2 D-Kiikku

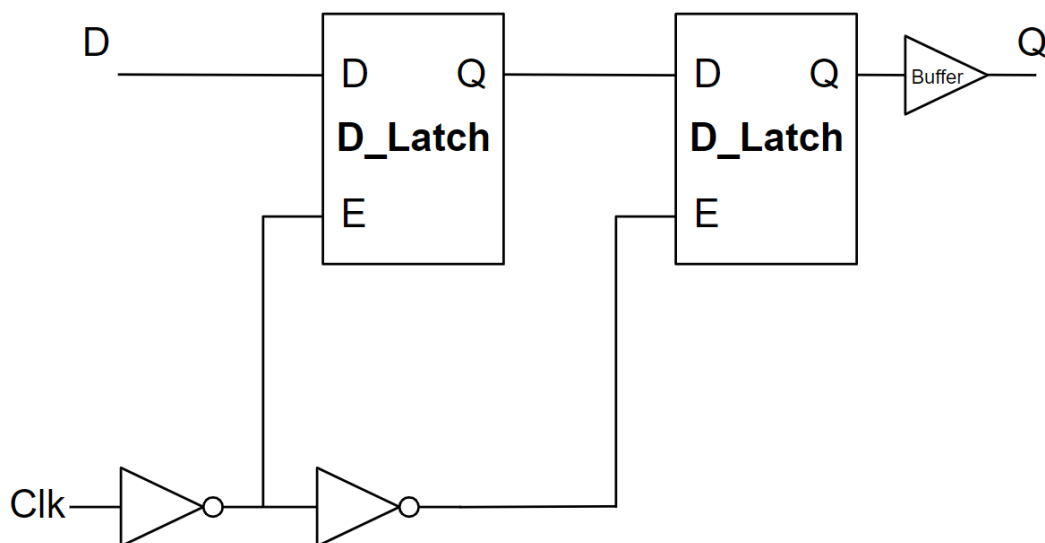
Kiikku on yksi digitaalilogiikan tärkeimpiä komponentteja ja sen pääasiallinen funktio on säilyttää tulossa oleva data kellojakson yli. Kiikussa on useimmiten kaksi tuloa, data ja kello, ja kellon reunalla kiikku vaihtaa tai säilyttää tilansa tulevan datan mukaan. Se siis seuraa tulosignaalia vain kellon tietyllä reunalla. On olemassa erilaisiin liipaisutyyleihin perustuvia kiikkuja, kuten esimerkiksi vastakkaisten reunojen liipaisuun perustuva kiikku (Data lock-out flip flop), reunaliipaistava kiikku (Edge-triggered flip flop), ja kiikku, jonka liipaisemiseen käytetään pulssia (Pulse triggered flip flop). Yleisimpiä kiikkutyppejä digitaalelektronikassa ovat SR-kiikku, D-kiikku, JK-kiikku ja T-kiikku. Tässä työssä rakennettiin yksituloinen D-kiikku, jonka toimintaperiaate näkyy taulukossa 2. [26]

Taulukko 2. D-kiikun totuustaulu [24, s.146-149].

D	Clk	Q
0	↑	0
1	↑	1

Kiikun rakentamisessa hyödynnettiin D-salpa, joka ohjausbitin ollessa 1 päästää tulon lähtöön ja ohjausbitin ollessa 0 säilyttää tilansa. Salpa itsessään perustuu kahteen 3-tila invertteriin ja niiden väliseen takaisinkytkentään. Kytkemällä salpoja kaksi peräkkäin ja kytkemällä niihin käänteiset ohjaukset, saadaan aikaan kiikku, joka vaihtaa tilaansa positiivisella kellon reunalla. Kuvassa 18 nähdään D-kiikun rakenne. Kiikun lähtö on puskuroitu sen takia, ettei kuorma vaikuttaisi niin paljon sen toimintaan. Käytetty rakenne myös tekee kiikun toiminnasta luotettavamman. Kellosignaalin kulkiessa inverttereiden läpi, voi käänteisellä kellolla olla viivettä suhteessa normaaliin kelloon, mikä aiheuttaa hieman vääristymää kellosignaaleihin (clock skew). Vääristymä näkyy siinä, että kellosignaalit voivat saada hetkellisesti samat arvot. Rakenteessa eivät signaalien päällekkäisyydet (0-0 ja 1-1) kuitenkaan aiheuta datasiignaalille hetkellistä suoraa reittiä lähtöön, jolloin tästä ei ole juurikaan haittaa.

Kun suunnitellaan kiikkuja, täytyy huomioida muutamia asioita. Kiikun toiminnan tulee olla nopeaa, mikä tarkoittaa mahdollisimman pientä viivettä tulon ja lähdön välillä, mahdollisimman lyhyttä positiivista setup-aikaa ja lyhyttä pitoaikaa. Koska kiikkuja tullaan käyttämään paljon ja melko tiheällä syklillä, täytyy tehonkulutus myös olla pientä. Sama kellosignaali ajaa useita kiikkuja, joten rakenteessa tulisi huomioida myös se, ettei kiikku kuormita kellosignaalia liikaa ja aiheuta siihen häiriöitä. Vastaavasti kiikun lähtö kytketään usein kauas tai se ajaa useita portteja, jolloin sen tulee kestää isoakin kapasitiivista kuormaa. Yleisellä tasollakin rakenteiden tulisi olla häiriönkestäviä siten, ettei tulossa tai käyttöjännitteissä olevat pienet häiriöt näy lähdössä tai vaikuta toimintaan.



Kuva 18. D-Kiikun lohkokaavio.

3.3.3 4-tuloinen Multiplekseri BIN/LIN dekodauksella

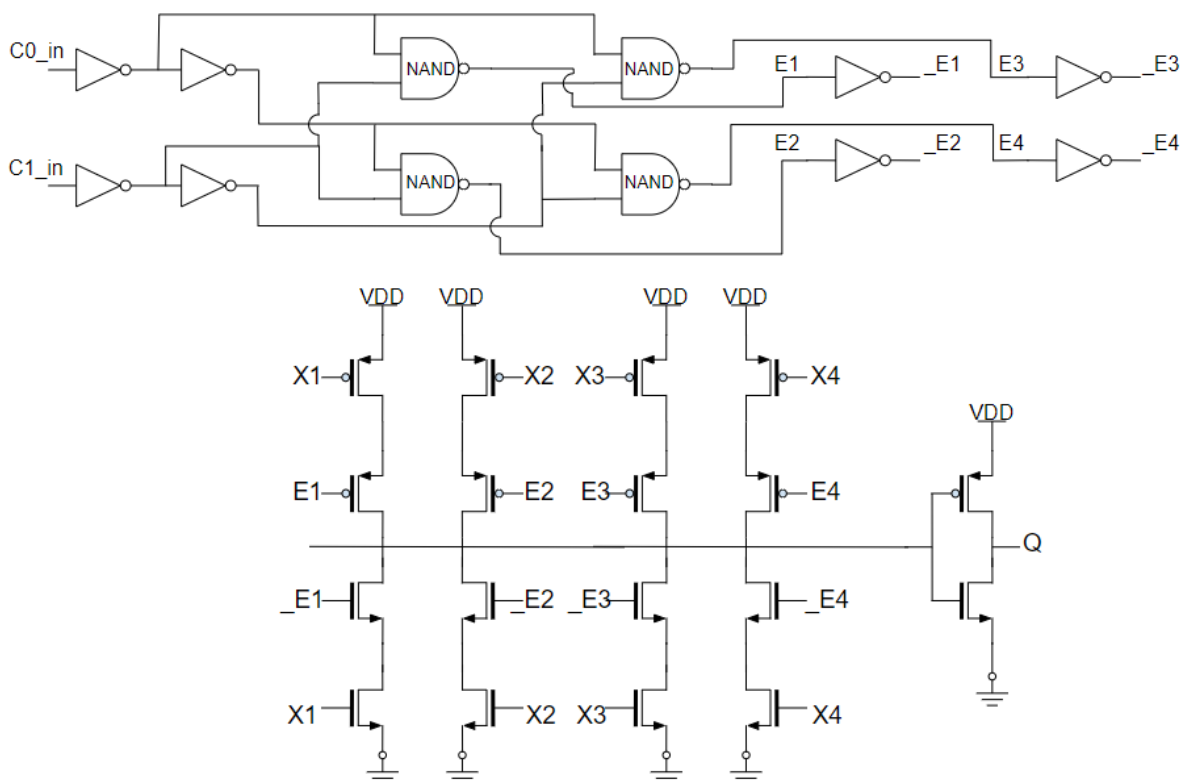
Multiplekseri on komponentti, jossa useasta tulosignaalistä yksi siirretään lähtöön tekemällä valinta ohjaussignaalien avulla. Tarvittavien ohjaussignaalien määrä on verrannollinen tulosignaalien määrään, esimerkiksi 8 tulosignaalistä voidaan valita yksi käyttäen kolmea bittä 000, 001, ..., 111. Komponentin avulla voidaan siis ohjata samaa paikkaa usealla eri signaalilla. Tämä komponentti rakennettiin käyttäen hyväksi BIN/LIN-dekoodausta, jossa kahdesta ohjaussignaalistä saatiin tarvittava määrä muunnoksia lähdön ohjelmoimiseksi. Normaalissa

rakenteessa ohjaussignaaleilla ohjataan suoraan lähtöä, mutta silloin transistoreja tulee useampi sarjaan, mikä hidastaa komponenttia huomattavasti. Lisäämällä rakenteeseen dekodauslogiikka kasvaa transistorien kokonaismäärä eli käytännössä pinta-ala, mutta komponentin nopeus paranee huomattavasti. Suunnittelussa joutuukin punnitsemaan eri ominaisuuksien välillä, mikä on hyödyllisintä missäkin tilanteessa. Logiikkakomponenttien täytyy olla nopeita, joten päädyttiin käyttämään hieman isompaa pinta-alaa. Multiplekserin lähtösignaali voidaan esittää seuraavasti Boolean lausekkeen avulla [26, s.118].

$$Y = X_1 \overline{C_1} \overline{C_0} + X_2 \overline{C_1} C_0 + X_3 C_1 \overline{C_0} + X_4 C_1 C_0.$$

Binary-Line dekodaus perustuu ajatukseen, jossa k kappaleesta tulosaaleja koodataan 2^k lähtösignaalia. Dekodaus suoritetaan käyttämällä kombinaatiologiikkaa, jolla saadaan haluttu vaste. Kuvassa 19 näkyvässä multiplekserin lohkokaaviossa on esitetty dekodaukseen käytetty kombinaatiologiikka. Tässä ohjaussignaaleista C_0 ja C_1 otettiin ensin käännteissignaalit, jotka kytkettiin yhdessä alkuperäisten signaalien kanssa NAND-portteihin siten, että saatiin kahdesta tulobitistä neljä ohjaussignaalia. Näiden neljän lähtösignaalin, sekä niiden käännteissignaalien avulla ohjattiin sitten 3-tila inverttereitä ja sitä kautta dataaaleista yksi lähtöön. Käytetty koodauslogiikka on siis 2-to-4, eli kahdesta ohjaussignaalista koodataan neljä. Ohjausbitit 00, 01, 10 ja 11 vastaavat valintoja X_1 , X_2 , X_3 ja X_4 .

Koodauslogiikassa käytetyt invertterirakenteet, sekä NAND-rakenteet on mitoitettu niitä suunniteltaessa ja tässä käytettiin alkuperäisten porttien vastaavia transistorikokoja. Lähtöpuolella käytettiin hyvin pieniä w/l-suhteita, jolloin saatiin piirikuviota varten tarpeeksi tilaa nostaa kaikki hilakontaktit PMOS- ja NMOS-transistorien välissä ylös.



Kuva 19. 4-tuloisen multiplekserin piirikaavio, jossa mukana lohkotason esitys BIN/LIN-dekodauslogiikasta.

3.3.4 Virta-DAC

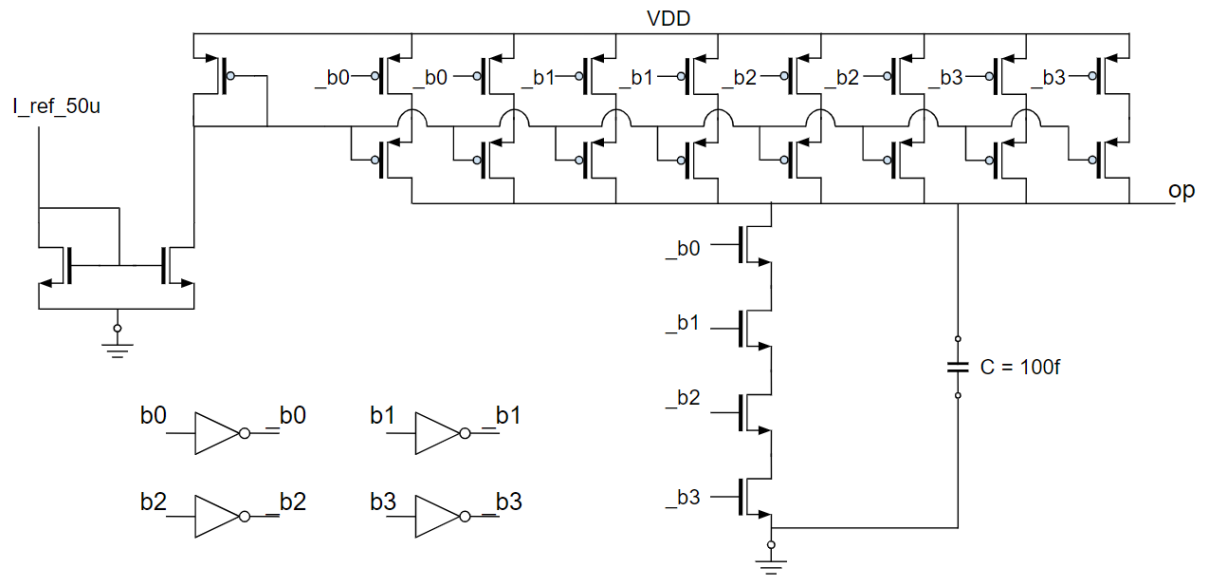
Virta-DAC:n pääasiallinen tehtävä on säätää lähtösolmun virtaa tulobittien funktiona. Virta-DAC on digitaali-analogia-muunnin, jossa lähtöjännitettä ohjataan ajamalla diodiksi kytketyn transistorin läpi virtaa ohjausbittien funktiona. Kuvassa 20 näkyvässä rakenteessa on 4-bittinen virta-DAC. Koska virta ajetaan käyttöjännitteestä PMOS-transistorien läpi kuormaan, tulee ohjausbitit kääntää, jotta saadaan transistori johtavaksi tulon ollessa looginen 1. Virta-DAC voi myös perustua NMOS-virtapeileihin [28]. Lohkon varsinainen toiminta perustuu siihen, että tuloon ajetaan $50 \mu\text{A}$ referenssivirtaa, jota peilataan NMOS-virtapeilillä lähtöpuolen virtapeiliin. Lähtöhaarassa puolestaan virtaa peilataan PMOS-virtapeileillä tulobittien mukaan tarvittava määrä lähtöön. Virtapeilissä peilaavan transistorin läpi menee referenssivirta I_1 , joka peilataan lähtösolmun virraksi I_2 seuraavalla suhteella

$$I_2 = \frac{S_2}{S_1} \cdot I_1, \quad (10)$$

missä S_1 ja S_2 ovat transistorien W/L-suhteet ja I_1 sekä I_2 ovat transistorien läpi kulkevat virrat. Mitoittamalla lähtöpuolen transistorin leveys esimerkiksi 2 kertaa isommaksi kuin peilaavan transistorin leveys, voidaan referenssivirta kaksinkertaistaa lähtöhaaraan. [17]

Lähdön ajavat virtapeilit on mitoitettu siten, että transistorien kokonaisleveys kasvaa bittisuhteessa ylöspäin. Ensimmäisen ohjausbitin ohjaamat virtapeilit ovat tietyn kokoisia, seuraavan bitin ohjaamat 2 kertaa suurempia, kolmannen 4 kertaa suurempia ja viimeiset ovat 8 kertaa suurempia. Näin saadaan lähtösolmun virtaa kasvatettua yhtä paljon jokaisella bittimuutoksella. Kun kaikki tulobitit ovat 0, ohjataan lähdön jännite alas. Tämä sen takia, ettei seuraavissa rakenteissa tule ongelmia virran tai kelluvan jännitteen kanssa, eli DAC on käytännössä suljettu. Diodikytketty transistori jätettiin piirikaavion ulkopuoliseksi komponentiksi, jolloin jokainen komponentin käyttäjä voi siirtää sen avulla lähtöjännitettä haluamaansa suuntaan. Näin komponentin käyttäminen ja jännitealueen säätö on mahdollisimman helppoa ja sen saa varmasti halutulle tasolle.

Virralla ajettavissa rakenteissa on tärkeä muistaa, että käyttöjännitteen ja maatason häiriöt näkyvät helposti suoraan lähdössä. Jos komponentti on päällä ja muualla piirillä vaihdetaan kytkinten tilaa toiseksi, voi transientin seurauksena lähtövirta heilahtaa ja aiheuttaa pahimmassa tapauksessa komponenttien hajoamista tai sammumista. Tällaiset äkilliset heilahtelut on hyvä suodattaa kondensaattorilla pois. Mitoittaessa suodatuskondensaattoria on hyvä huomioida se, että kondensaattorin koko on komponentin kokoon verrattuna todella suuri. Kapasitanssin kasvattamiseksi päädyttiin ratkaisuun, jossa käytetään metalleja aina tasolle 7 asti, jolloin saatiin kondensaattorin (MOM, Metal-Oxide-Metal) pinta-alaa pienennettyä järkevälle tasolle. Haluttu suodatuskapasitanssin arvo oli 100 fF , mikä 3 alinta metallikerrosta käyttäen olisi tarkoittanut monta kertaa varsinaisen komponentin kokoista pinta-alaa. Johdotuksen kannalta useiden metallitasojen käyttäminen on hankalaa ja suuri koko vaatii myös leveämmät johtimet piiriltä kondensaattorille.



Kuva 20. Virta-DAC piirikaavio.

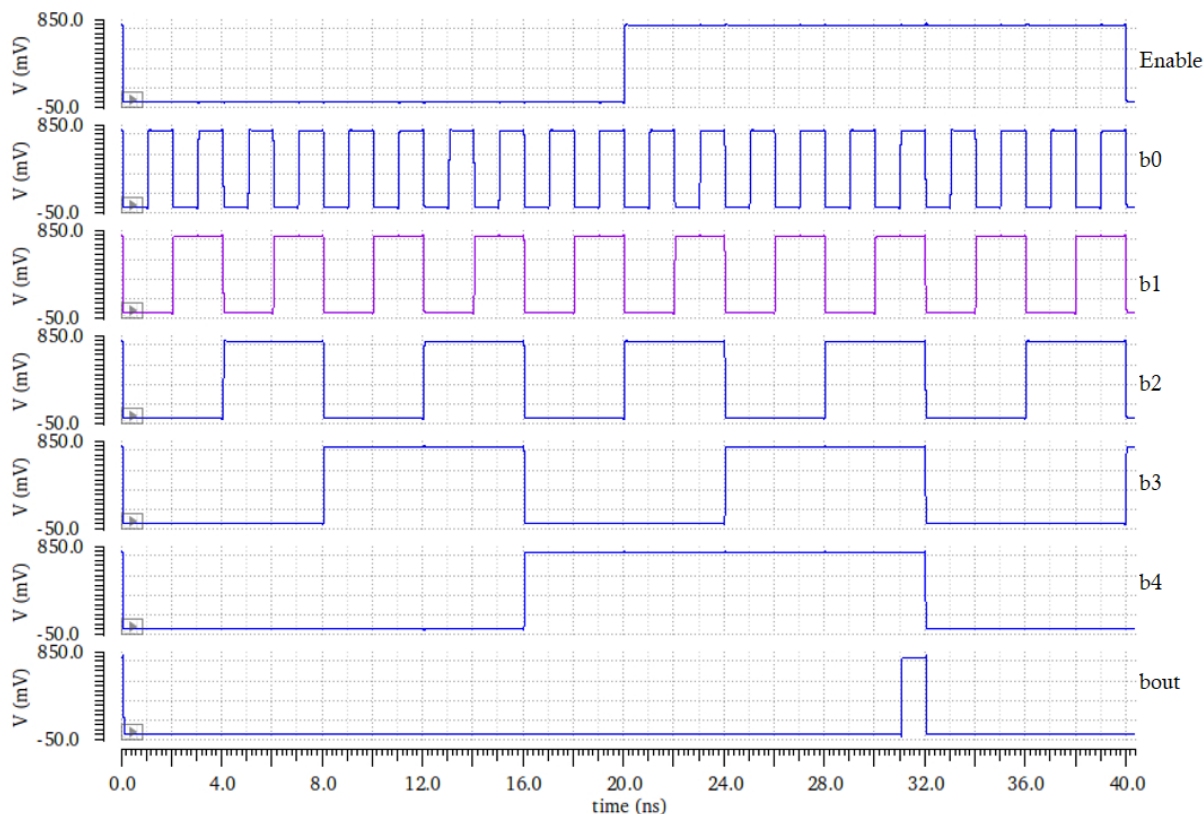
4 PIIRIRAKENTEIDEN SIMULOINTI

Piirikaavioiden mitoitus ja simuloinnit suoritettiin yhtä aikaa. Usein simuloinnin jälkeen jouduttiin vähän muuttamaan transistorien kokoja tai muita arvoja, jotta saatiin vaste muuttumaan haluttuun suuntaan. Lähtökohtina mitoitukseen ja komponenttien suunnitteluun käytetyt periaatteet nopeudesta, pienestä koosta ja symmetrisistä nousu- ja laskuajoista ohjasivat myös simulointeja paljon.

4.1 Rakenteiden toiminta

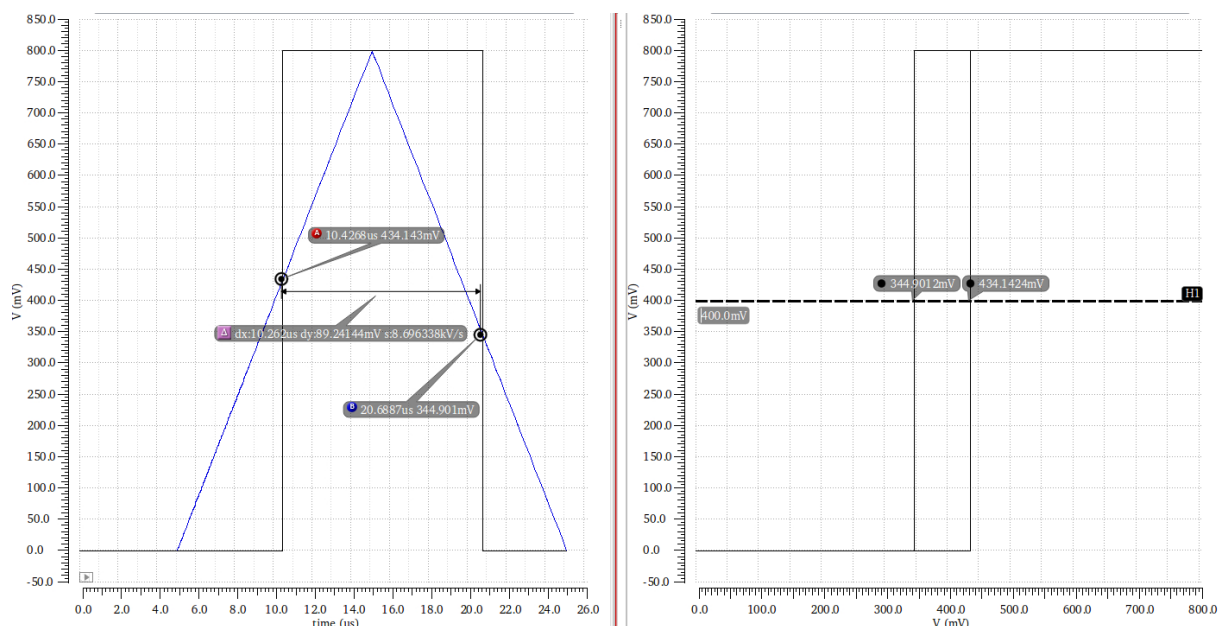
Ennen varsinaista mitoitusprosessia tehtiin rakenteille toimintasimulointi, jossa ajettiin tuloportteihin toimintalogiikkaa vastaavat signaalit sisään ja varmistettiin, että lähtö noudattaa haluttua logiikkaa. Tässä vaiheessa useimmiten tulosignaalien nousu- tai laskuajoihin ei kiinnitetty niin paljon huomiota vaan signaalit generoitiin ideaalisesti. Toimintasimuloinnissa ei tarvittu nopeita signaaleja vaan riitti että logiikkatasot ovat oikeanlaiset ja käyttöjännitteet halutulla tasolla. Toimintasimuloinnissa verrattiin tuloksia kunkin komponentin totuustauluun, mistä näki suoraan seuraako lähtö oikeaa logiikkaa.

Yksinkertaisten komponenttien, kuten NAND-, NOR- ja AND-porttien (kts. kuvat 7, 8, 14 ja 15) toiminnan simulointi oli todella suoraviivaista. Riippuen tulobittien määrästä täytyi tuloon saada luotua kaikki erilaiset kombinaatiot, joilla porttia tullaan mahdollisesti ajamaan ja näin varmistaa se, että mikään bittiyhdistelmä ei aiheuta komponentin toimintaan häiriöitä. Kuvassa 21 nähdään 5-tuloisen salpaohjauksen sisältävän AND-portin toiminnan simulointitulokset. Lähtösignaali bout on 1 vain silloin kun kaikki 5 tulosignaalia b0-b4 ovat 1, sekä sallintesignaali Enable on päällä. Tässä tapauksessa jouduttiin käymään $2^5 = 32$ erilaista bittikombinaatiota läpi.



Kuva 21. Salpa-ohjaussignaalilla varustetun 5-tuloisen AND-portin simulointitulokset.

Mutkikkaammissa komponenteissa toimintaa varmennettiin myös mitoituksen yhteydessä. Tästä esimerkkinä on schmitt-triggeri (kts. kuva 12), jonka kynnysjännitteet piti saada asetettua siten, että kynnykset ovat noin käyttöjännitteen kymmenesosan päässä toisistaan lähellä jännitetasojen puoliväliä. Tässä mitoituksessa transistorien koko oli ratkaiseva tekijä toiminnan varmentamiseen. Mitoitus suoritettiin ajamalla hidas kolmioaalto sisään ja simuloimalla lähtöjännite tulojännitteen funktiona. Tuloksena saatiin kuvan 22 mukainen hystereesiskäyrä, jonka leveys on 90 mV, eli noin 11% käyttöjännitteestä. Kynnykset, joissa komponentti vaihtaa tilaansa, ovat nousevalle signaalille 434 mV ja laskevalle 345 mV, joten tulossignaali kestää olla lähes 90 mV kohinaa mukana. Kuvasta nähdään myös helposti, että schmitt-triggeriä voidaan käyttää muuttamaan hitaita nousu- ja laskuaikoja terävämmiksi. Tämä on hyödyllinen ominaisuus esimerkiksi komponenteissa, joissa kellon reunojen tulee olla terävät.



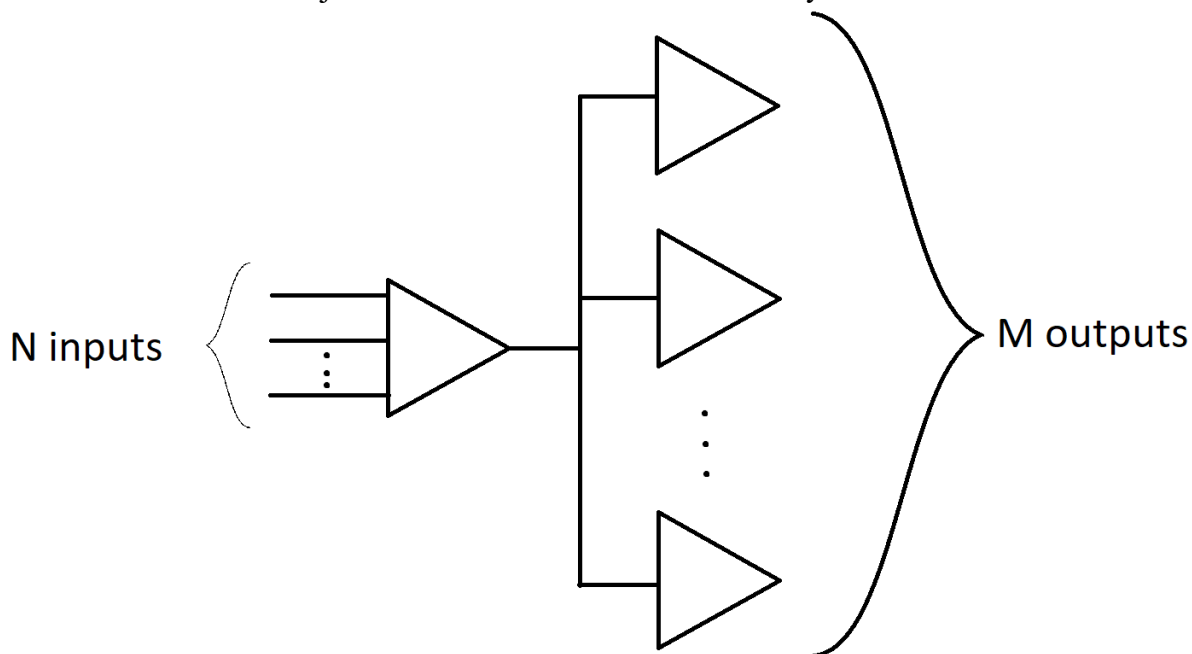
Kuva 22. Ei-kääntävän schmitt-triggerin simulointitulos, vasemmalla transientti simulointi ja oikealla hystereesiskäyrä.

4.2 Kuorma ja toimintaympäristö

Tärkeä osa simulointeja on se, missä ympäristössä ja millaisilla signaaleilla piirejä ohjataan. Simulointien tulisi vastata mahdollisimman hyvin käytännön toimintaympäristöä. Ensimmäisiä asioita, joita täytyy huomioida, ovat tulossignaalin muoto sekä nousu- ja laskuajat. Niiden tulee vastata mahdollisimman hyvin todellisia tulossignaaleja. Tämän vuoksi testipenkeissä käytettyjä signaaleja hieman pyöristettiin ja muotoiltiin ajamalla ne esimerkiksi inverttereiden läpi. Signaaligeneraattorin nousu- ja laskuajat säädettiin lähelle todellista, noin 5 ps aikaa. Generaattorin luomaa signaalia saatettiin myös hieman viivästyä ajamalla se 100 Ω sarjavastuksen läpi. Muita asioita, jotka vaikuttavat rakenteiden mitoitukseen, puskuroinnin tarpeeseen ja toimintaan ovat käyttöjännitteet, komponenttien virrat, kapasitiivinen kuormitus eli käytännössä lähtösolmuun kytkettyjen komponenttien määrä ja suodatuksen tarpeellisuus.

4.2.1 Fan-in ja Fan-out

Termillä Fan-in tarkoitetaan sitä, kuinka monta tuloporttia komponentissa on. Fan-out puolestaan tarkoittaa sitä, kuinka montaa porttia komponentin lähtö ajaa. Näistä merkityksellisempi on fanout, koska se on suoraan kytköksissä kapasitiivisen kuorman määrään. Termit fan-out ja fan-in on havainnollistettuna esitetty kuvassa 23.



Kuva 23. Fan-in ja Fan-out havainnollistettuna.

Digitaalilogiikan suunnittelussa usein joudutaan pohtimaan sitä, mikä on optimaalisin tapa mitoittaa komponentti ja minkä kokoisella kuormalla sen viiveen ja muut ominaisuudet saisi parhaimmaksi. Tässä on päädytty tulokseen, jossa optimaalisin tulos saadaan silloin, kun mitoitettavalla portilla neljää identtistä porttia lähtösolmussa (FO4 – Fan-out of Four). Alkuperäinen ongelma oli mitoittaa suurehkoa kapasitiivista kuormaa C_L ajava N mittainen invertteriketju siten, että viive tulosta lähtöön minimoituisi. Ketjun ensimmäinen invertteri on minimikokoinen ja koot kasvavat asteittain loppua kohti. Ongelman ratkaisussa päädyttiin tulokseen, jossa jokaisen ketjussa olevan invertterin koko edelliseen verrattuna kasvaa sen verran, että seuraavan portin tulokapasitanssi vastaa noin neljää ajavan invertterin tulokapasitanssia. Tällöin jokaisen asteen efektiivinen fan-out on lähellä 4:ää. Tämä tuottaa minimiviiveen ketjun tulosta lähtöön ja näin ollen on optimaalinen ratkaisu. Tarkempi kuvaus ongelmasta ja sen tulosten johtaminen löytyy lähteestä [3, s207-211].

Tässä työssä päädyttiin simulointien avulla melkein samaan tulokseen. Tilastollisesti digitaalilogiikkapiireissä keskimääräinen porttien kuormitus on noin 3-5 invertteriä, joten simuloinneissa päätettiin käyttää aina 5 invertteriä kuormana. Niissä tapauksissa, joissa on ollut tarve ajaa suurempaa kuormaa, on simulointeihin lisätty useampia portteja. Lisäksi virta- ja tehonkulutusmittausten ohessa simulointeja suoritettiin myös ilman kuormaa eli vain komponentin oman lähtökapasitanssin kanssa.

4.2.2 Kapasitanssit, resistanssit ja viive

Piirien kapasitanssi koostuu useasta eri lähteestä. Transistoreissa on hilan, nielun ja lähteen välillä kapasitiivinen kytkös johtuen käytetyistä metallitasoista ja niiden välissä olevista eristävästä materiaaleista. Johtimet aiheuttavat merkittävän osan kapasitansseista sillä niissä yleensä tulee verrattain isoja pinta-aloja eri metallikerrosten väliin. Kapasitanssia on oikeastaan piirien kaikkien johtimien ja metallikerrosten välillä, joiden välissä on eristävää materiaalia. Simuloinneissa tärkeimmät kapasitanssit, jotka tulee huomioida, ovat johtimista aiheutuvat kapasitanssit, komponentin oma lähtökapasitanssi sekä lähtöön kytkettyjen komponenttien tulokapasitanssit. Kaikkien näiden yhteisvaikutus näkyy suoraan komponentin lähtösignaalin ominaisuuksissa.

Piireissä johtavat materiaalit aiheuttavat myös resistanssia. Lähtösolmussa näkyvä kokonaisresistanssi tulee transistorien resistansseista ja johtimien sekä epäideaalisuuksien tuomista resistansseista. Johtimien resistanssi määräytyy materiaalin resistiivisyyden, johtimen leveyden ja pituuden mukaan.

Useimmat piirit voidaan ajatella ensimmäisen asteen RC-piireinä, eli niissä on pääosin transistorin kanavasta aiheutuva resistanssi, jonka läpi kondensaattoria varataan ja puretaan. RC-piirille lähtöjännite voidaan laskea kaavasta (1). Jos yhtälöön sijoitetaan logiikkatasojen puoliväli $V_{out} = 1/2 \cdot VDD$, voidaan yhtälöstä ratkaista viive

$$\frac{1}{2} \cdot VDD = VDD \cdot \left(1 - e^{-\frac{t}{\tau}}\right), \quad (11)$$

josta ratkaistaan t ja saadaan

$$t = \ln(2) \cdot \tau = RC_L \cdot \ln(2), \quad (12)$$

missä R on portin kokonaisresistanssi ja C_L portin kokonaiskapasitanssi. [2, s.29-30] Yhtälöstä nähdään, että kapasitanssin kasvaessa piiri hidastuu, eli viivettä tulee lähdön asettumiseen lisää. Tämän vuoksi kapasitiivinen kuormitus portissa tulee pitää järkevällä tasolla. Yhtälössä oleva resistanssi tulee PMOS- ja NMOS-transistorien ominaisuuksista. Silloin kun transistori johtaa, sen resistanssi on luokkaa 10 k Ω - 100 k Ω , kun johtamattomana se on noin 10 M Ω - 100M Ω .

Kuvassa 24 näkyy minimikokoisen invertterin virta- ja jännitekäyrät 10 fF kuormakapasitanssilla. Kuvan perusteella voidaan laskea arviot NMOS ja PMOS transistoreiden resistansseista johtavassa tilassa ja ei-johtavassa tilassa kaavoilla

$$R_{on} = \frac{U_{on}}{I_{on}}, \quad (13)$$

$$R_{off} = \frac{U_{off}}{I_{off}}, \quad (14)$$

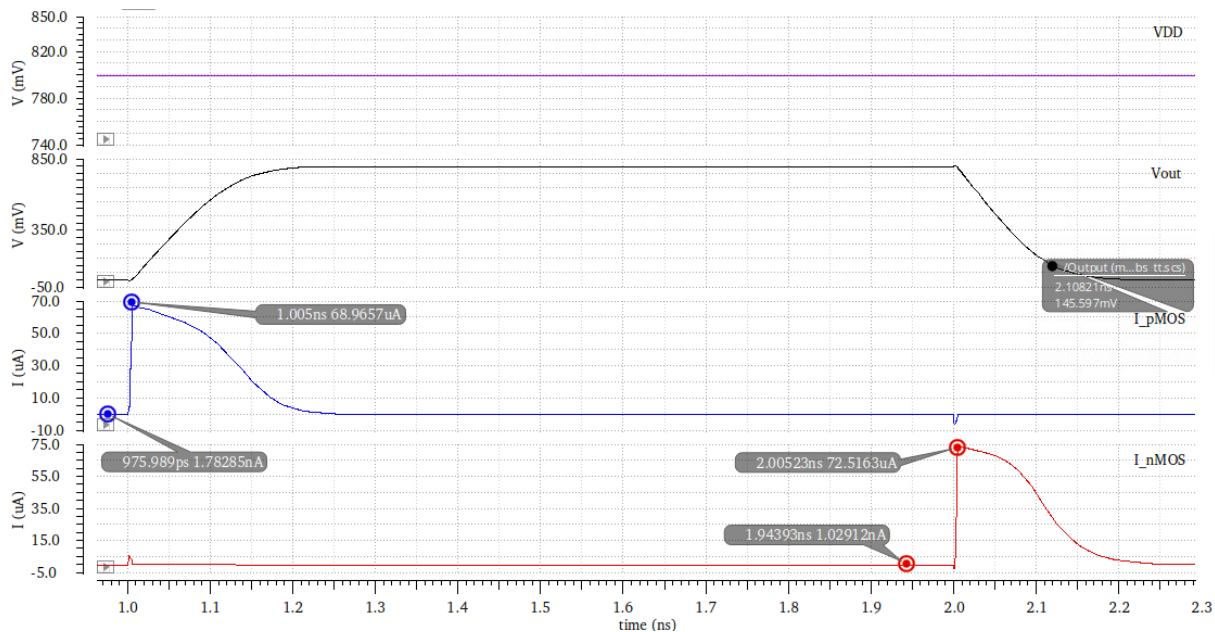
missä R on resistanssi, U on jännite transistorin yli ja I on transistorin virta. Sijoittamalla kuvan 24 päällä-virrat ja -jännitteet kaavaan (13) ja kiinni-virrat ja -jännitteet kaavaan (14) saadaan laskettua resistanssit

$$R_{on,PMOS} = \frac{800 \text{ mV}}{69 \mu\text{A}} = 11,59 \text{ k}\Omega, \quad (15)$$

$$R_{off,PMOS} = \frac{800 \text{ mV}}{1.8 \text{ nA}} = 4,44 \text{ M}\Omega, \quad (16)$$

$$R_{on,NMOS} = \frac{800 \text{ mV}}{73 \text{ }\mu\text{A}} = 10,96 \text{ k}\Omega, \quad (17)$$

$$R_{off,NMOS} = \frac{800 \text{ mV}}{1.0 \text{ nA}} = 800 \text{ M}\Omega. \quad (18)$$



Kuva 24. N- ja PMOS transistorien jännitteet ja virrat minimikokoisessa invertterissä 10 fF kuormakapasitanssilla.

Korkeilla taajuuksilla toimivista piireistä tehdään lähes aina simulointimalli, joka sisältää piirikuvion johdotuksista ja rakenteista tulevat kapasitanssit, hajakapasitanssit, johtimien induktanssit, resistanssit ja muut ekstraktoidut parasittiset komponentit. Taajuuden kasvaessa myös häiriöitä alkaa näkyä monella tapaa. Yleensä jos piiriin tuodaan RF-signaali, sen voi mitata lähes kaikista pinneistä. Tämä johtuu siitä, että korkeataajuisille signaaleille kapasitiivinen kytkeytyminen tarkoittaa lähes oikosulkua, jolloin signaali näkyy melkein joka paikassa. Tämän kirjaston komponenttien simuloinneissa ei tarvittu karakterisoituja malleja kuin kaikista kriittisimmille komponenteille, kuten virta-DAC ja virtapeilit. Näissä rakenteissa täytyy varmistaa, ettei johtimista ja muista kytkennöistä tule liikaa ominaisuuksia heikentäviä vaikutuksia. Rakenteiden kapasitanssia voi arvioida käyttämällä levykondensaattorin kapasitanssin kaavaa

$$C = \frac{\epsilon_0 \epsilon_r A}{d}, \quad (19)$$

missä C on kapasitanssi, ϵ_0 on tyhjiön permittiivisyys, ϵ_r on eristemateriaalin suhteellinen permittiivisyys, A on kondensaattorin pinta-ala ja d on metallien välinen etäisyys.

Taulukossa 3 on esitetty viivemittausten sekä nousu- ja laskuaikojen mittaustuloksia melkein kaikista työssä suunnitelluista komponenteista. Suunnittelun tavoitteena oli saada aikaan komponentteja, joiden nousu- ja laskuajat lähtösignaalilla ovat samaa suuruusluokkaa ja joiden viiveet ovat mahdollisimman pieniä. Taulukosta nähdään, että suurimmaksi osaksi

jokaisen komponentin nousu- ja laskuajat ovat lähellä toisiaan, mutta AND-porttien laskuajat ovat huomattavasti lyhempiä kuin nousuajat lukuun ottamatta 5-tuloista AND-porttia, joka antaa hyvin symmetrisen tuloksen. Pienin invertterirakenne on tehty siten, että siinä on vain yksi PMOS- ja NMOS-transistoripari, jotka koostuvat yhdestä hilasta. Pienimmän invertterin suhteellinen kapasitiivinen kuorma on isompi kuin muilla, joten se on hitaampi. Simuloinneissa kaikkien porttien kuormana on käytetty 5 minimikokoista invertteriä eli simuloinneissa FO = 5.

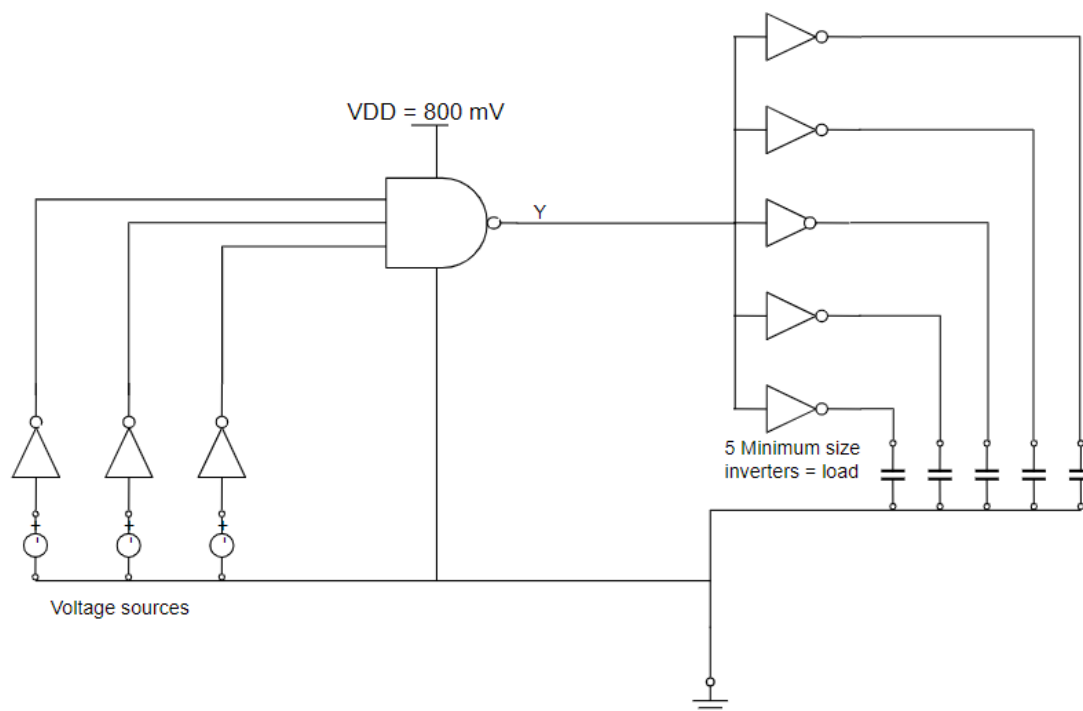
Taulukko 3. Simulointitulokset komponenttien viiveistä, kun FO = 5

Komponentti	Viive [ps]	Nousu-/Laskuaika lähtösignaalille [ps]/[ps]
Nand_X2	23,82	12,65/11,65
Nand_X3	30,38	8,69/8,09
Nor_X2	21,69	5,86/4,90
Nor_X3	30,30	6,18/5,13
Mux_2_1	24,79	11,87/10,37
Mux_4_1_BIN/LIN	33,76	12,92/11,57
Schmitt_trigger	18,78	9,40/8,21
DFF	34,40	24,03/20,59
DLatch	18,09	5,01/5,00
And_X2	22,58	11,09/7,41
And_X3	28,46	17,62/9,70
And_X5_LE	51,29	4,87/4,58
Inv_X3	13,62	20,45/17,29
Inv_X9	6,40	7,42/6,57
Inv_X12	5,54	5,91/5,30

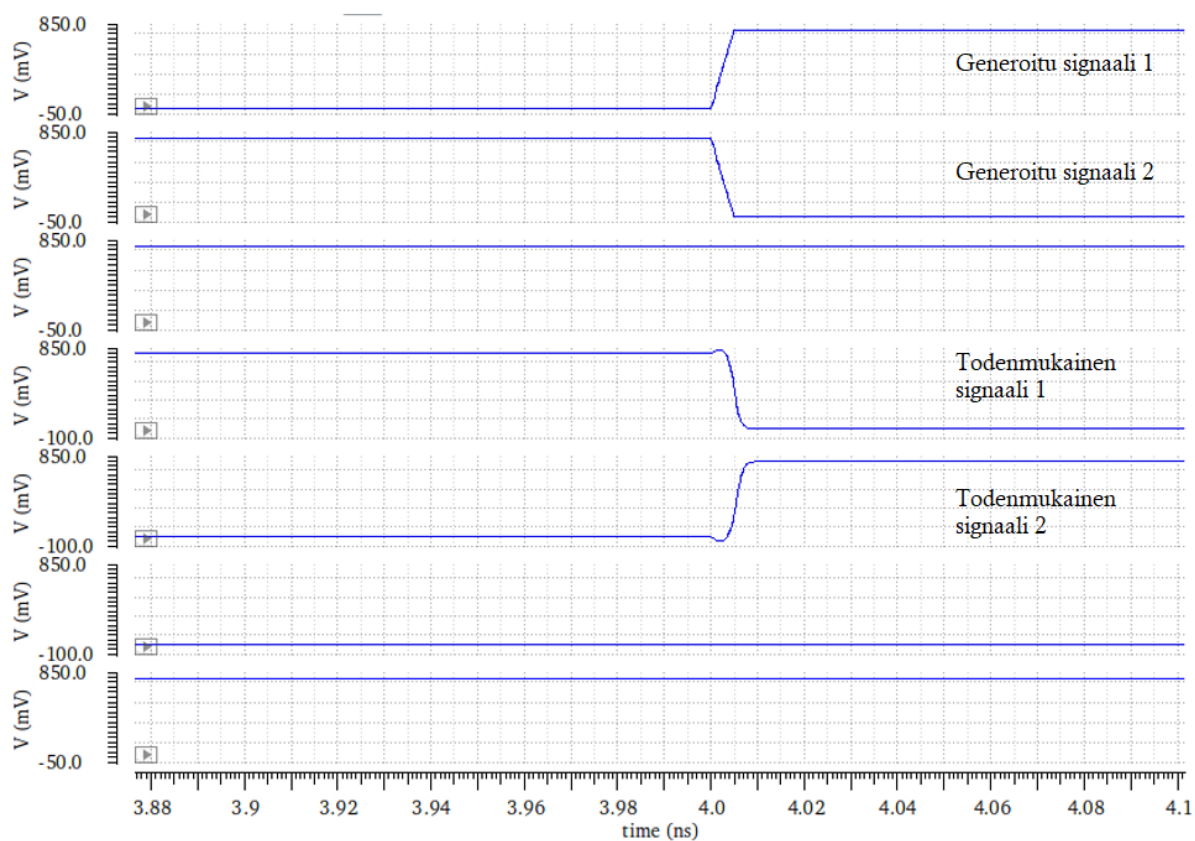
4.3 Testipenkit ja simulointi

Simuloinneissa käytetyt testipenkit ovat suunnittelun luotettavuuden kannalta olennaisin osa. Se, että simuloidaan oikeanlaisilla ja todellisuutta vastaavilla signaaleilla sekä realistisilla kuormituksilla osoittaa komponentin toimivuuden käytännössä. Signaaleiden nopeudet, taajuudet ja jännitteet täytyy olla myös oikein asetettu simulointeja varten.

Kuvassa 25 on esitetty 3-tuloisen NAND-portin yksinkertaistettu testipenkki. Suurin osa simuloinneista on suunniteltu vastaavalla tavalla. Signaalilähteet luovat noin 5 ps nousu- ja laskuajoilla olevia signaaleja, joiden jakson mitta on säädetty siten, että kolmeen tuloporttiin saadaan kaikki mahdolliset bittikombinaatiot. Koska kyseessä on hyvin pienen viivanleveyden teknologia, eivät transistorit kestä kovin isoja jännitteitä, joten sekä logiikkatasot, että käyttöjännite ovat maksimissaan noin 800 mV. Alempi logiikkataso on tässä tapauksessa maa eli 0 V. Signaalit pyritään generoimaan mahdollisimman realistisiksi, mutta todellisuudessa signaalit eivät koskaan ole teräväkulmaisia, joten niiden nurkkia muokataan vastaamaan enemmän todellisuutta ajamalla ne minimikokoisen invertterin läpi. Tämän voi nähdä hyvin kuvasta 26, jossa kaksi ylintä käyrää ovat jännitelähteillä generoidut teräväreunaiset signaalit ja kaksi keskimmäistä ovat todellista vastaavia, joissa on mukana kapasitanssista aiheutuvaa jännitemuutosta ja pyöristymistä. Jälkimmäisissä myös nousu- ja laskuajat vastaavat ajavan signaalin ominaisuuksia. Joissakin rakenteissa käytettiin lisänä sarjavastusta viivästämään signaaleja hieman. NAND-portin lähtösolmussa kuormana on 5 kpl minimikokoista invertteriä, mikä vastaa aika hyvin keskimääräistä kuormaa logiikkasoluille. Tällä kuormalla piiri on lähellä todellista toimintaympäristöä.



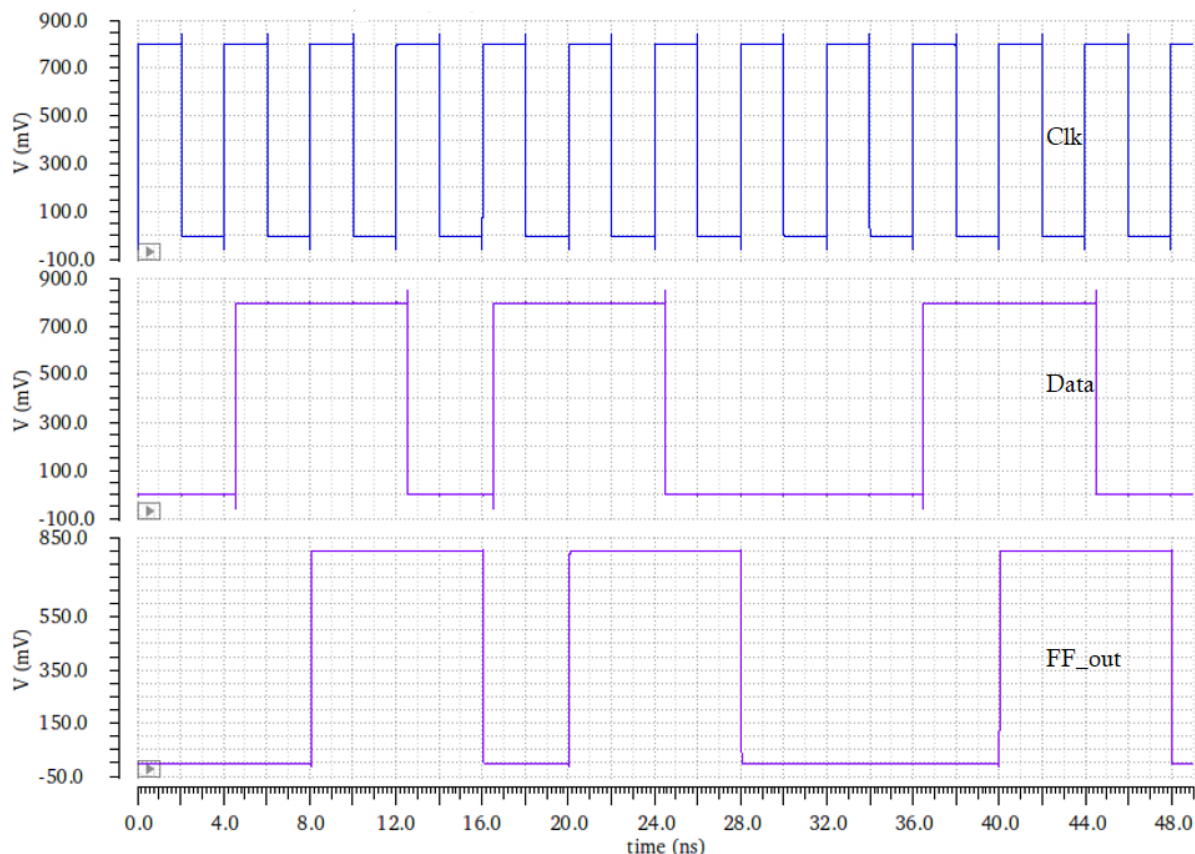
Kuva 25. NAND_X3 simuloinnin testipenkin piirikaavio.



Kuva 26. Generoidun signaalin ja todenmukaisen signaalin vertailua.

4.4 D-Kiikun simulointitulokset

Kuvassa 27 nähdään D-kiikun (kts. kuva 18) toiminnan simulointitulokset. Kellosignaali on ylimpänä ja kellon nousevalla reunalla datasignaalin arvo luetaan lähtöön. Kuvasta näkyy myös se, että tulosignaalien pienet heilahdukset transienttien vaikutuksesta eivät päädy lähtöön, vaan lähtösignaali on puhdas. Tämä johtuu siitä, etteivät tulosignaalien pienet heilahdukset ylitä tai alita logiikkatasojen puoliväliä, jolloin ne eivät vaikuta lähtösignaaliin. Myös lähtösolmun kapasitiivinen kuorma omalta osaltaan siistii signaalia.



Kuva 27. D-Kiikun toimintasimuloinnin tulokset.

4.4.1 Viiveen, setup- ja pito-ajan mittaukset

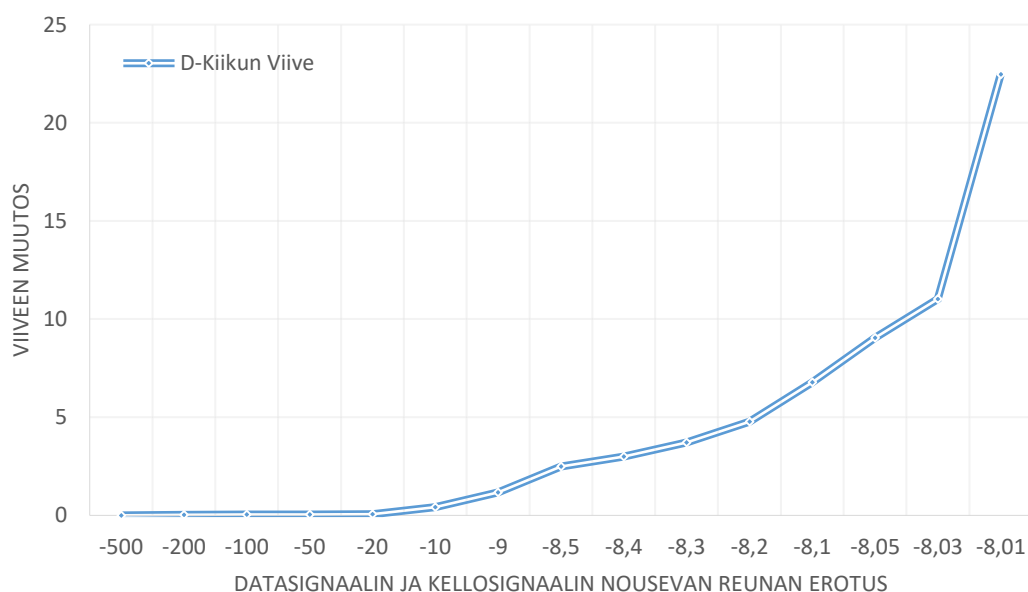
Kuten jo aiemmassa kappaleessa mainittiin, on D-kiikun tärkeimpiä ominaisuuksia viive ja setup-aika, eli aika, joka vaaditaan datan ja kellosignaalin väliin toiminnan varmistamiseksi. Taulukossa 4 on esitetty kiikun setup-ajan mittaustulokset ja kuvassa 28 näkyy viiveen kasvu tulosignaalin saapumisajan ja kellosignaalin saapumisajan erotuksen funktiona. Taulukosta nähdään, että kiikun viive sen toimiessa oikein on 34,40 ps, eli vähintään tämä aika tarvitaan kellosignaalin nousemisesta rekisterin lähdön asettumiseen. Kuvasta puolestaan nähdään, että datan saapuminen 9 ps ennen kelloa ei juurikaan vaikuta viiveeseen, mutta eron pienentyessä kohti 8 ps aiheuttaa viiveen lähtemisen kasvuun eksponentiaalisesti. Tästä voidaan päätellä, että kiikun setup-aika on noin 9 ps. Kuvaajan 0-taso tarkoittaa sitä, että viiveen muutos on lähellä nollaa, jolloin itse viive on lähellä alkuperäistä 34,40 ps viive arvoa.

Kiikku myös päästää datan läpi vielä 8 ps erolla, mikä tarkoittaa sitä, että jos siirtorekisterissä on kellosignaalin reitillä noin 1 ps mittainen johdin, voi data mennä kahden kiikun läpi yhdellä kellon nousevalla reunalla, koska kellosignaali viivästyy juuri ratkaisevan

mittaisen ajanhetken. Tämä ei aiheuta ongelmaa esimerkiksi tilakoneissa, joissa on kiikkujen välissä sen verran logiikkaa, että signaalilla kestää aina yli 1 ps kulkea kiikkujen välillä. Siirtorekistereissä ja osoitedekoodereissa ei tällaista saa tapahtua. Tästä aiheutuvia ongelmia voidaan välttää esimerkiksi juoksuttamalla siirtorekistereissä data ja kello eri suuntiin, jolloin kellosignaalin nouseva reuna ei tule väärällä ajanhetkellä rekisterin kiikuille. Tällaisia ongelmia ei todennäköisesti kuitenkaan näillä komponenteilla ilmene, sillä kiikun oma viive on yli 30 ps, joten datan viive poikkeaa reilusti johtimen aiheuttamasta viiveestä kellosignaalin reitillä.

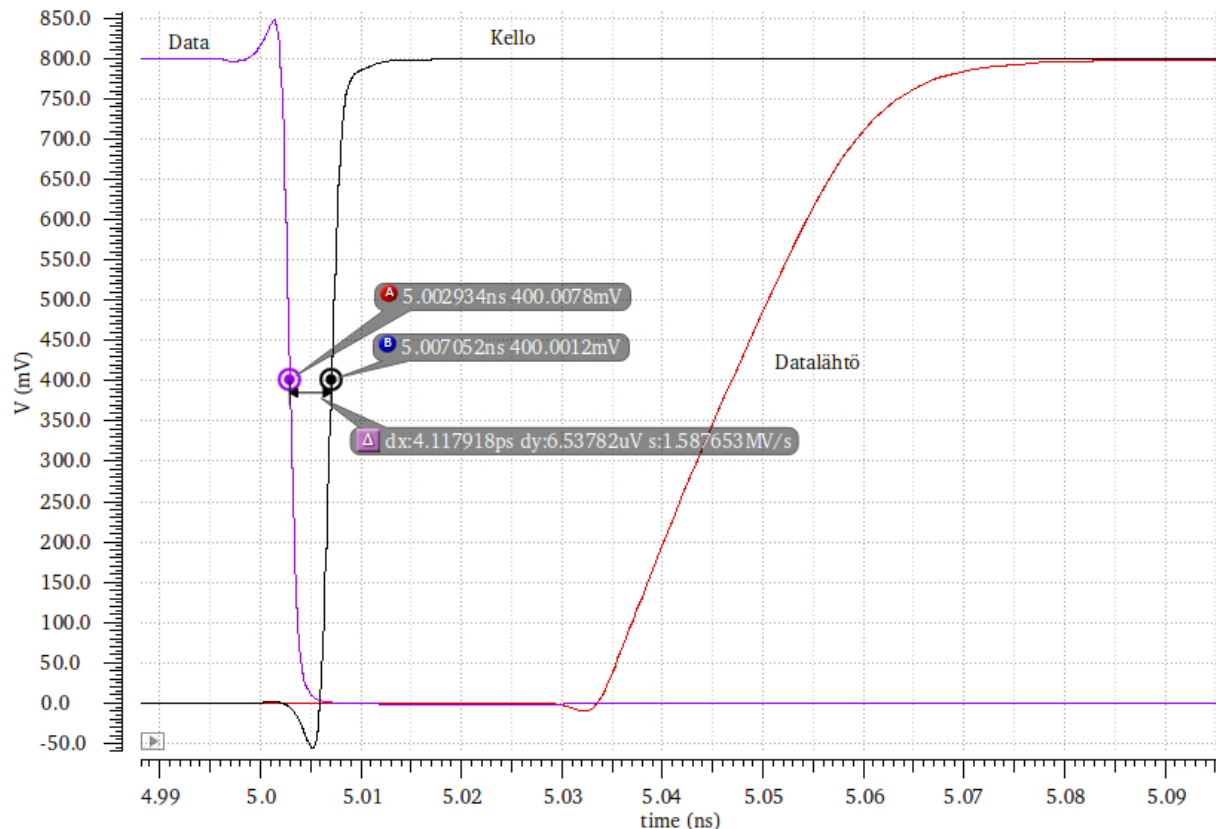
Taulukko 4. D-Kiikun viiveen ja setup-ajan mittaustulokset.

Data [ps]	Kello [ps]	Ero [ps]	Toimiiko?	Viive [ps]	Viive muutos [ps]
500	1000	-500	Kyllä	34,40	0,00
800	1000	-200	Kyllä	34,42	0,02
900	1000	-100	Kyllä	34,44	0,04
950	1000	-50	Kyllä	34,45	0,05
980	1000	-20	Kyllä	34,46	0,06
990	1000	-10	Kyllä	34,81	0,41
991	1000	-9	Kyllä	35,57	1,17
991,5	1000	-8,5	Kyllä	36,89	2,49
991,6	1000	-8,4	Kyllä	37,39	2,99
991,7	1000	-8,3	Kyllä	38,12	3,72
991,8	1000	-8,2	Kyllä	39,18	4,78
991,9	1000	-8,1	Kyllä	41,18	6,78
991,95	1000	-8,05	Kyllä	43,44	9,04
991,97	1000	-8,03	Kyllä	45,42	11,02
991,99	1000	-8,01	Kyllä	56,86	22,46
992	1000	-8	Ei	-	-
993	1000	-7	Ei	-	-
1000	1000	0	Ei	-	-



Kuva 28. Viiveen muutos datan ja kellosignaalin erotuksen suhteessa.

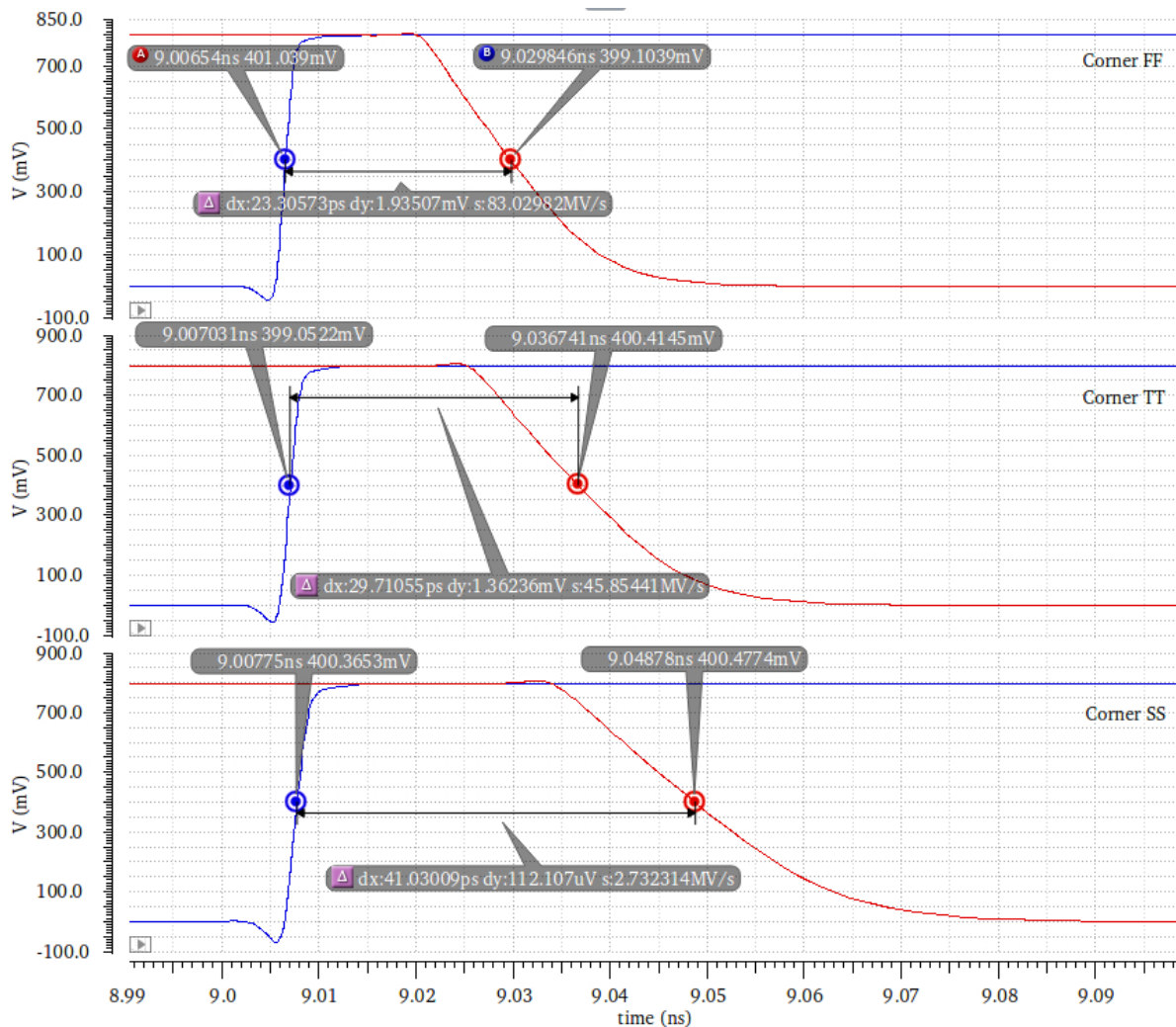
Kiikun simuloinneissa mitattiin myös kiikun pito-aika, eli aika, jonka datan täytyy olla ylhäällä kellon nousevan reunan jälkeen. Simuloinneissa ilmeni, että simuloitavalle rakenteelle pito-aika on niukasti negatiivinen, -4 ps, mikä johtuu siitä, että data ja kello kulkevat hiukan eri reittiä. Datasignaalin tullessa muutamaa pikosekuntia aiemmin tuloportteihin, ovat ne kellosignaalin kanssa yhtä aikaa perillä rakenteen sisällä. Data kulkee 3-tila invertterin läpi ja kello puolestaan kahden invertterin läpi, jolloin niiden viiveet reitillä eroavat toisistaan hieman. Kuvassa 30 nähdään pito-ajan simulointitulokset, kun data tulee noin 4 ps ennen kellosignaalin nousevaa reunaa ja lähtö edelleen nousee ylös.



Kuva 29 D-Kiikun pito-ajan simulointitulokset.

4.4.2 Prosessi- ja lämpötilavaihteluiden vaikutus viiveeseen

Kuvassa 29 on simuloitu kiikun viive prosessinvaihtelujen ja lämpötilan ääripäissä. Ylimmäinen kuvaaja vastaa nurkkaa FF (Fast-Fast) lämpötilassa -15°C, keskimäinen kuvaaja vastaa nurkkaa TT (Typical-Typical) lämpötilassa 27°C ja alimmainen kuva vastaa nurkkaa SS (Slow-Slow) lämpötilassa 85°C. Simuloinneissa olisi voitu käydä prosessinvaihtelut ja lämpötilan vaihtelut paljon tarkemminkin läpi simuloimalla myös ääripäiden välistä, mutta ääripään tilanteet antavat kuvan parhaimmasta mahdollisesta sekä huonoimmasta mahdollisesta tilanteesta. Jos ääripäiden simuloinnit osoittavat, että piiri toimii järkevissä rajoissa, voidaan tehdä oletus, että piiri toimii myös ääripäiden välisissä lämpötiloissa ja prosessinvaihteluissa. Kuvan perusteella nähdään, että viive lähdön laskevan reunan osalta on minimissään 23,3 ps FF-simuloinnissa ja maksimissaan 41,0 ps SS-simuloinnissa, mikä kertoo sen, että lämpötila ja prosessinvaihtelut eivät aiheuta mahdollottoman suurta haittaa kiikun toimintaan.

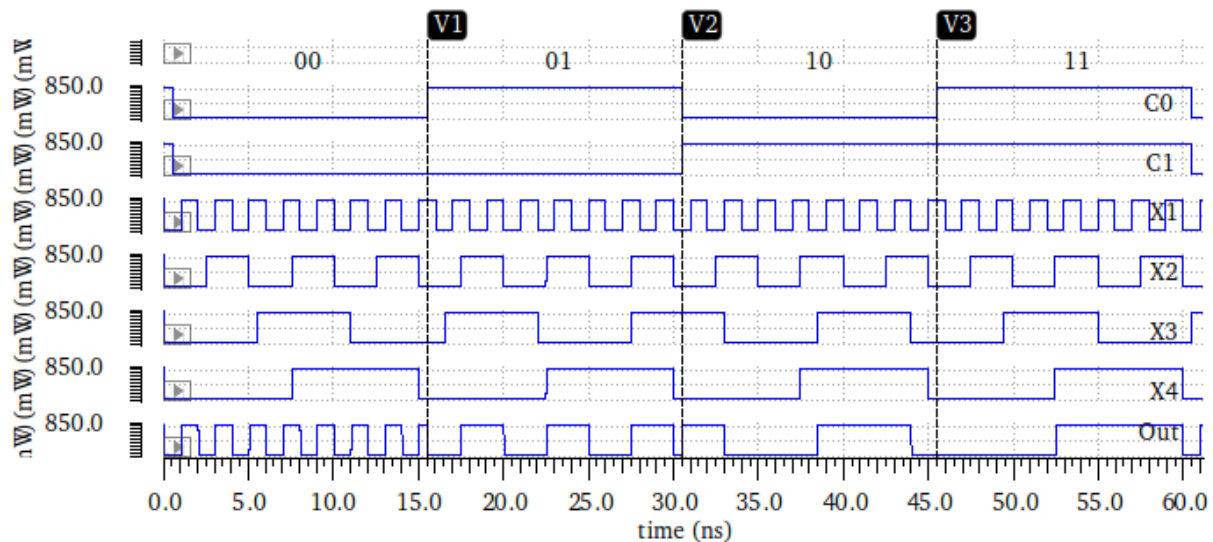


Kuva 30. Prosessi- ja lämpötilavaihteluiden vaikutus D-kiikun viiveeseen.

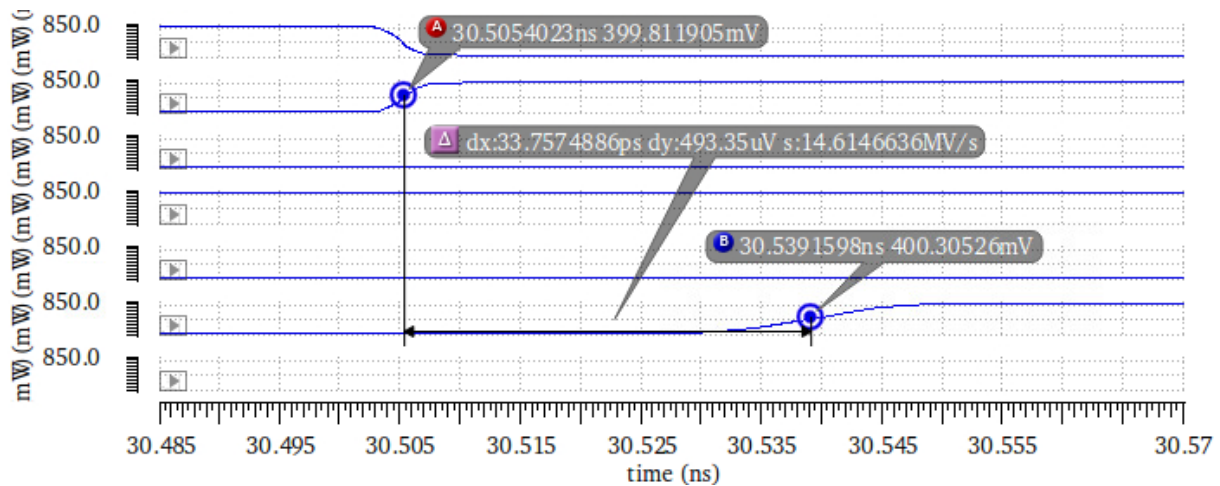
4.5 4-tuloisen multiplekserin simulointitulokset

Multiplekserin (kts. kuva 19) toiminnassa on olennaista se, että ohjaussignaalien avulla valitaan tulosignaaleista yksi signaali lähtöön. Kuvassa 31 nähdään 4-tuloisen multiplekserin ohjaussignaalit ja lähtösignaali kaikissa eri tilanteissa. Ohjausbittien C1 ja C0 mahdolliset kombinaatiot 00, 01, 10 ja 11, on merkitty kuvaan. Niiden perusteella valitaan tuloista ohjausbittejä vastaava signaali lähtösolmuun. Kun tulossa on 00, valitaan X1-tulosignaali lähtöön ja bittien kasvaessa myös lähtösignaali vaihtuu aina seuraavaan, kunnes lopulta viimeinen tulosignaali X4 valitaan biteillä 11. Lähtösignaalista voidaan selkeästi havaita, että kukin tulosignaali näkyy komponentin lähdössä ohjausbittien funktiona. Kuvaan on merkitty pystyviivat V1 – V3 niille ajanhetkille, joissa ohjausbitit vaihtavat tilaansa.

Viiveen simulointitulokset 4-tuloiselle multiplekserille on esitetty kuvassa 32. Viive on mitattu kohdasta, jossa ohjausbittien muutos aiheuttaa lähtösignaalin muutoksen hetkellä, jossa valittava tulosignaali on eri toisessa logiikkatasossa kuin sen hetkinen tulosignaali. Näin saadaan viiveeksi ohjaussignaalin muutoksesta lähtöön 33,76 ps, mikä on samaa suuruusluokkaa kuin kiikun viive.



Kuva 31. Multiplexerin toiminnan simulointitulokset.



Kuva 32. Multiplexerin viiveen simulointitulos.

4.6 Virta DAC simulointitulokset

Kuvassa 33 nähdään virta-DAC:n simuloinnissa käytetty testipenkki, jossa on myös lähtöjännitteen asettamiseen käytetty diodikytketty transistori. Transistorin kokoa säätämällä voidaan siirtää jännitealuetta haluttuun suuntaan. Kasvattamalla transistorin kokoa saadaan aluetta laskettua ja pienentämällä nostettua, sillä transistoriin ajettavan virran määrä kasvaa bittien funktiona ja isompi transistori pystyy ajamaan enemmän virtaa läpi pienemmällä V_{DS} jännitteellä, kun taas pieni transistori vaatii isomman jännitteen isommalle virralle. Kuvan lähtösignaali yhdistettiin ohjattavaan piiriin ison vastuksen kautta, jotta välttyttäisiin mahdollisilta häiriöiltä ja DAC syöttämä virta menisi kokonaan diodikytketyn transistorin läpi.

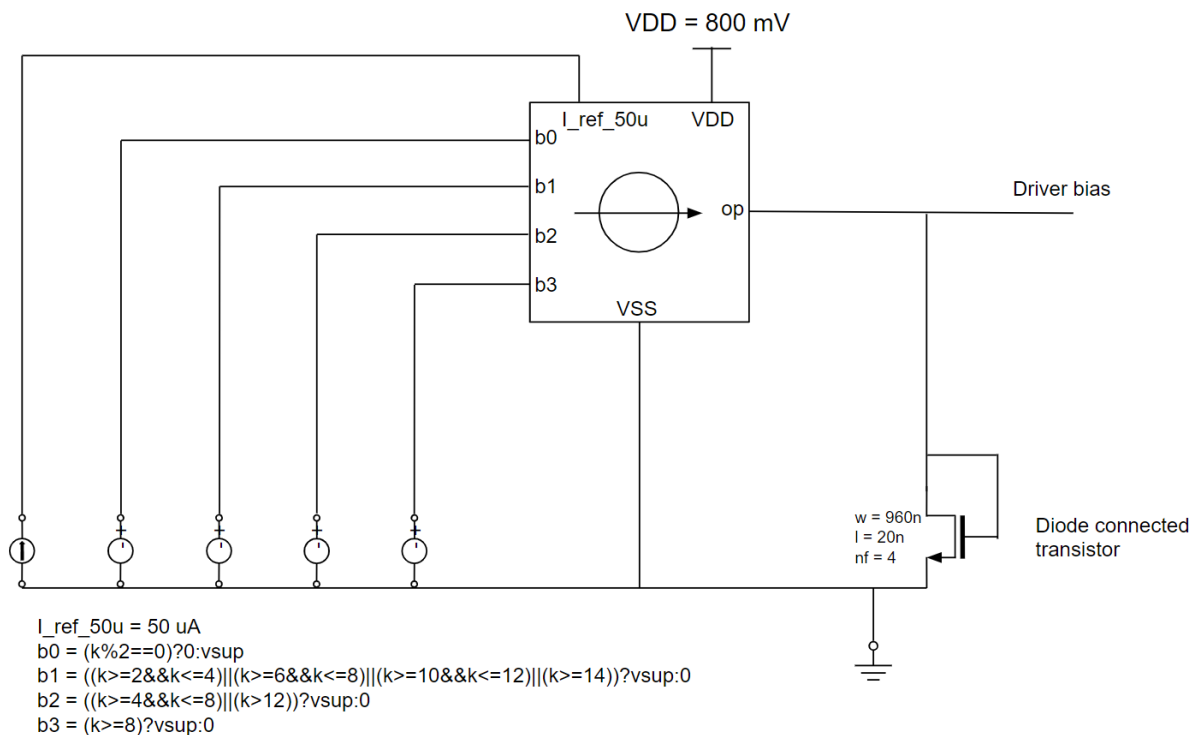
Simuloinneissa hyödynnettiin bittien ohjelmoinnissa lineaarista muuttujan pyyhkäisyä, jossa muuttujaa k pyyhkäistiin arvosta 0 arvoon 16 ja ohjausbitit ohjelmoitiin esittämään muuttujan sen hetkinen numeroarvo 4:llä bitillä. Näin saatiin pyyhkäistyä bitit arvosta 0000 \rightarrow 1111 ja käytyä kaikki mahdolliset ohjaussignaalit läpi. Ehtolauseissa jännitelähteiden arvo asetettiin käyttöjännitteeseen, jos lausekkeen ehto täyttyi ja vastaavasti maahan, jos ehto on epätosi. Kunkin bitin ehtolauseet olivat seuraavanlaiset

```

bit0 = (k%2==0)?0:vsup,
bit1 = ((k>=2&& k<=4)|| (k>=6&& k<=8)|| (k>=10&& k<=12)|| (k>=14))?vsup:0,
bit2 = ((k>=4&& k<=8)|| (k>12))?vsup:0,
bit3 = (k>=8)?vsup:0,

```

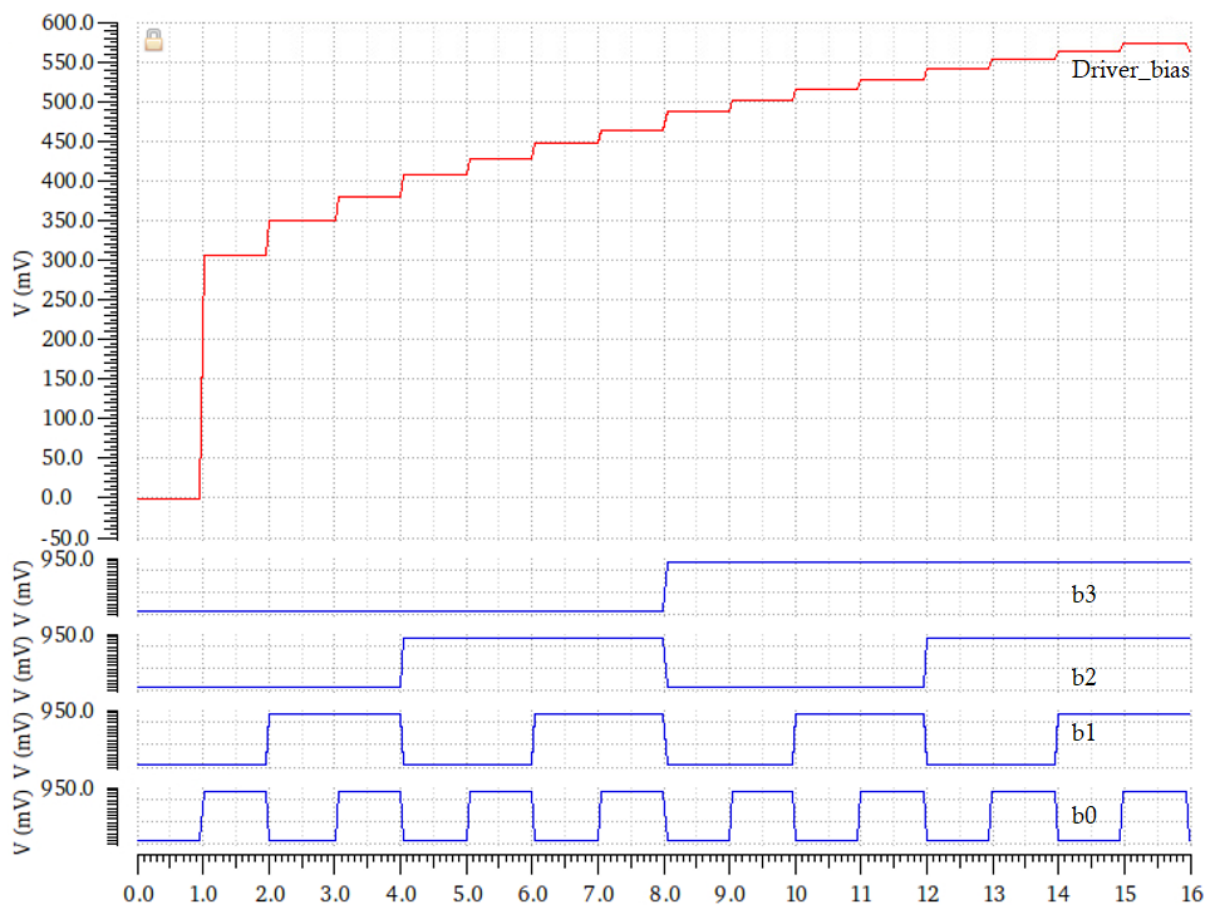
missä kunkin ehtolauseen rakenne koostuu osista *muuttuja = ehtolauseke?tosi:epätosi*, eli jos ehtolause on tosi, kaksoispisteen vasemmalla puolella oleva arvo sijoitetaan muuttujaan ja jos epätosi, niin oikealla puolella oleva sijoitetaan. Ehtolause voi sisältää useita eri ehtoja, joiden täytyy joko toteutua yhtä aikaa (&&, ja-operaattori) tai jonkun niistä täytyy toteutua (||, tai-operaattori). Simulointiympäristön ehtolauseet noudattavat cadencen oman SKILL-kielen syntaksia. Suurin osa operaatioista ja operaattoreista on SKILL-kielessä samoja kuin yleisesti käytössä olevat, mutta jotkut bittiooperaatiot ja suurin osa komennoista on omia, Cadencen ohjelmistoon liittyviä. Tässä esimerkiksi bit0-muuttuja asetetaan nolaksi, jos jakojäännös jaettaessa k 2:lla on nolla, muutoin asetetaan bit0-muuttujan arvoksi vsup-muuttuja eli käyttöjännite. Simuloinneissa bittimuutosten nousuajalla ei ollut niin isoa merkitystä, sillä simulointia ei tehty transienttiallyysinä, vaan dc-analyysinä. Tällöin voidaan piirtää lähtösignaali tulobittien funktiona ja kuvaajassa näkyvä signaalin nousuaika on riippuvainen muuttujan k pyyhkäisyyn askelvälistä. Mitä pienempi askelväli, sitä lyhempi nousuaika kuvaajassa on. Simuloinneissa käytettiin k:n pyyhkäisyyn 0,1 askelväliä, minkä voi nähdä kuvaajasta.



Kuva 33. Virta-DAC:n simulointiin käytetty testipenkki.

4.6.1 Virta DAC jännitealue

Lähtöjännitteen haluttu asettumisalue on noin 250 mV – 550 mV eli hyvin tyypillinen CMOS transistorin kynnysjännitteen ympäristössä oleva jännite. Digitaalinen ohjaus varmistaa sen, että jännitettä voidaan säätää hyvin helposti tällä alueella ja näin ollen tutkia esimerkiksi transistorin käyttäytymistä kynnysjännitteen lähettyvillä. Kun kaikki tulobitit ovat nollia, on lähtö kytkettynä maatasoon, jotta seuraavalle komponentille ei koituisi mitään haittaa. Kuvan 34 lähtöjännitteen käyrästä näemme, että mitoitettu jännitealue on 306 mV – 575 mV. Kuvan käyrästä nähdään myös se, että jänniteaskelmien koko pienenee ylöspäin mentäessä, mikä johtuu diodiksi kytketyn transistorin virtaominaisuuksista. Diodiksi kytketyllä transistorilla $V_{GS} = V_{DS}$, jolloin laite on saturaatioalueella, koska $V_{DS} > V_{GS} - V_{th}$. Käyrän alkupuolella askelmien koko on noin 45 mV ja viimeisten jänniteaskelmien koko on vain 15 mV eli kolmannes alkuperäisestä.

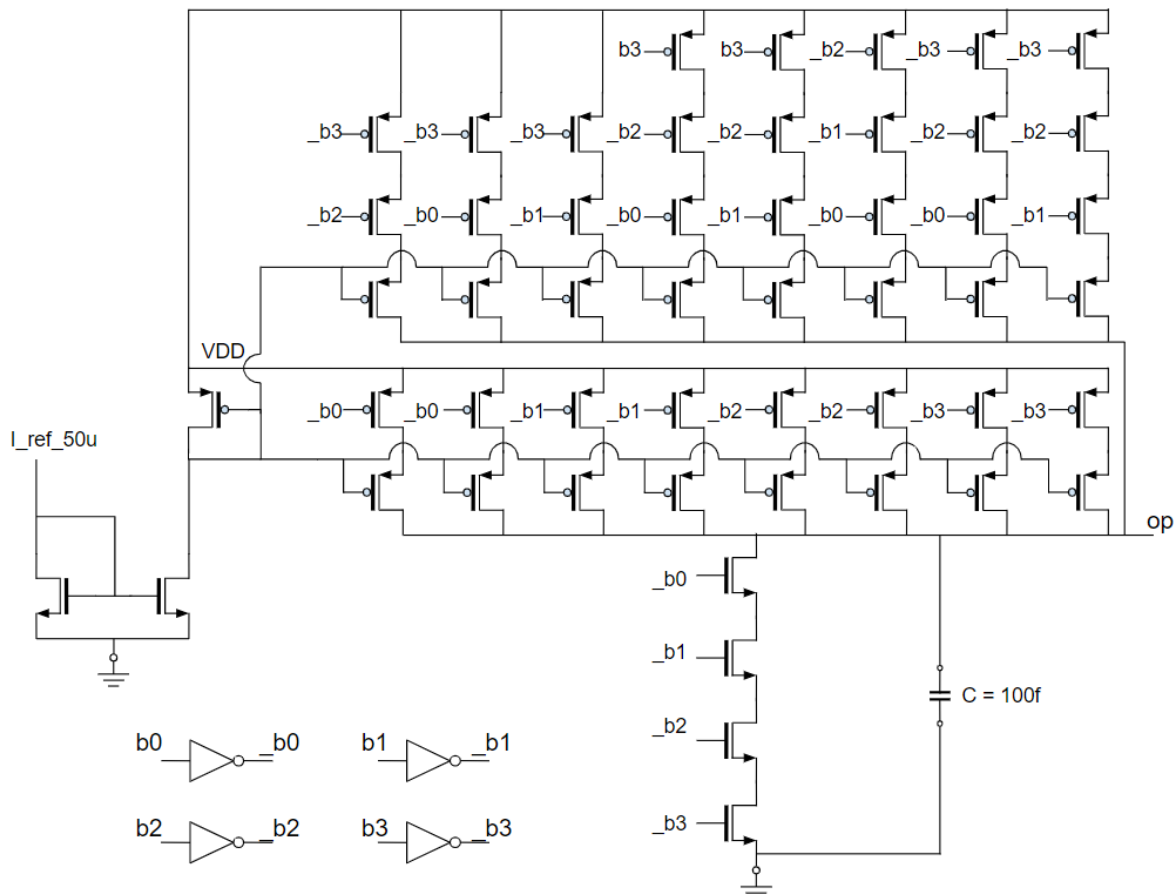


Kuva 34. Virta-DAC lähtöjännitekäyrä ohjausbittien funktiona.

4.6.2 Virta DAC jännitealueen linearisointi

Jos muuntimen lähtöjännitteen muutos diodikuormalla halutaan lineaariseksi, yksi mahdollinen korjauskeino virta-DAC:in jänniteaskelmiin on lisätä loppua kohti virransyöttöä isommaksi, tällöin saadaan kuvan 36 mukainen jännitekäyrä, missä askelmien koko pysyy jatkuvasti 25 mV – 35 mV suuruisena. Lähtöjännitealue kasvoi myös noin 100 mV verran, mikä voidaan havaita myös kuvasta. Jännitealueen minimi siirtyi hieman alemmas arvoon 297 mV ja ohjauksella tuotettava maksimi siirtyi ylemmäs arvoon 691 mV. Lähtöjännitteen linearisoimiseksi

piirirakenteeseen lisättiin virtapeilejä, jotka kytkeytyvät eri bittisanojen kohdalla päälle ja lisäävät kyseisen ohjauksen virtaa. Muokattu rakenne on esitetty kuvassa 35.



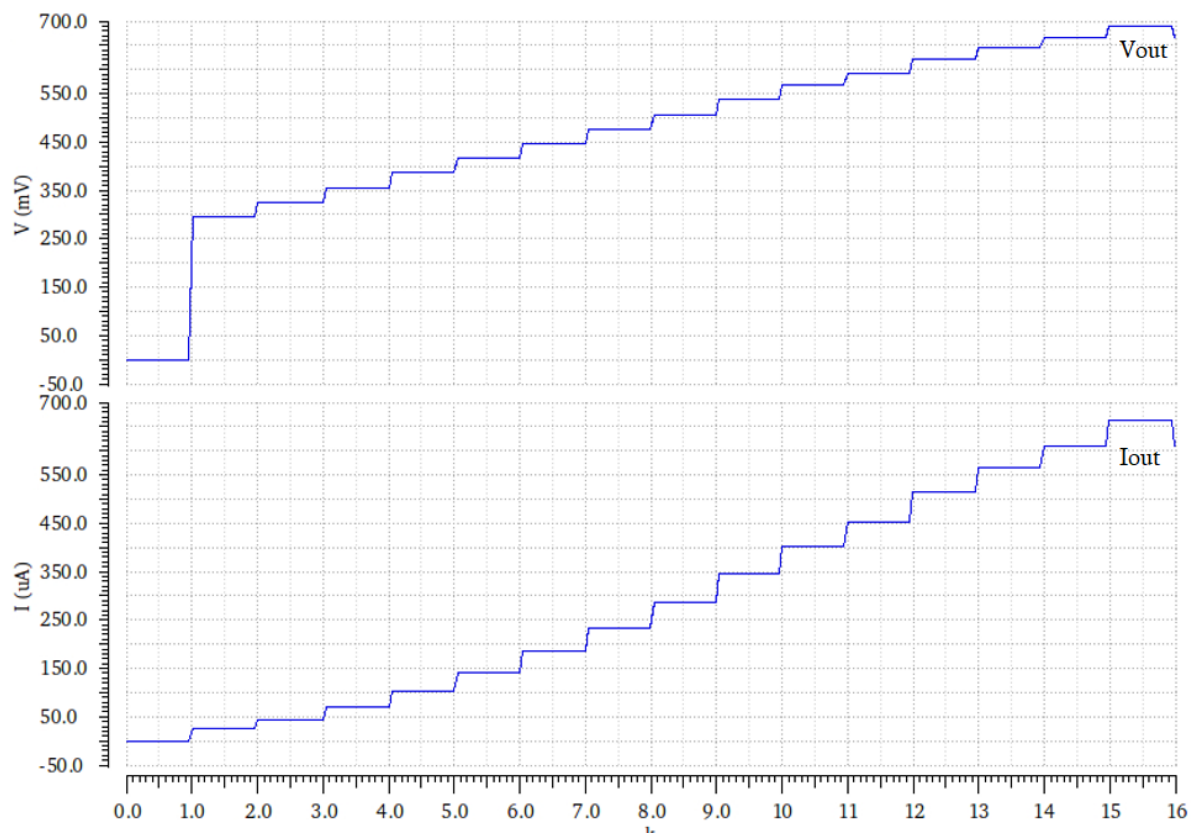
Kuva 35. Virta-DAC, johon on lisätty virtapeilejä nostamaan virtaa isompien bittisanojen kohdalla.

Kuvassa 36 voidaan havaita, että virransyöttö kasvaa selvästi k:n saadessa isoja arvoja. Maksimivirta kasvoi linearisoinnissa aina 664 μA suuruuteen, mikä tuo puolestaan ongelmia komponentin kestävyudessa, sillä tämä on hyvin lähellä teknologian tuomia virtarajoituksia. Alkuperäinen virta-DAC syöttää maksimissaan 414 μA virtaa kuvassa 33 näkyvän diodikytketyn transistorin läpi. Virransyötön lisääminen toi myös virta-DAC piirikaavioon noin 30 transistoria lisää ja kaksinkertaistaa sen pinta-alan. Todennäköisesti näiden virtojen suuruusluokassa täytyisi transistorimallia vaihtaa tehokkaampaan ja enemmän virtaa kestäväan, lisäksi piirikuvion johdotuksesta tulisi paljon leveämpiä. Simulointien perusteella nähdään, että transistorit ovat hyvin epälineaarisesti toimivia. Jos virtaa kasvatetaan lineaarisesti, mikä yleensä on haluttu vaste, jännitteen nousu hidastuu loppua kohden ja vastaavasti jännitteen noustessa lineaarisesti, virrankulutus lähtee kasvamaan neliöllisesti. Tämä johtuu diodikytketyn transistorin luonteesta ja voidaan nähdä virtayhtälöstä [3, s.93]

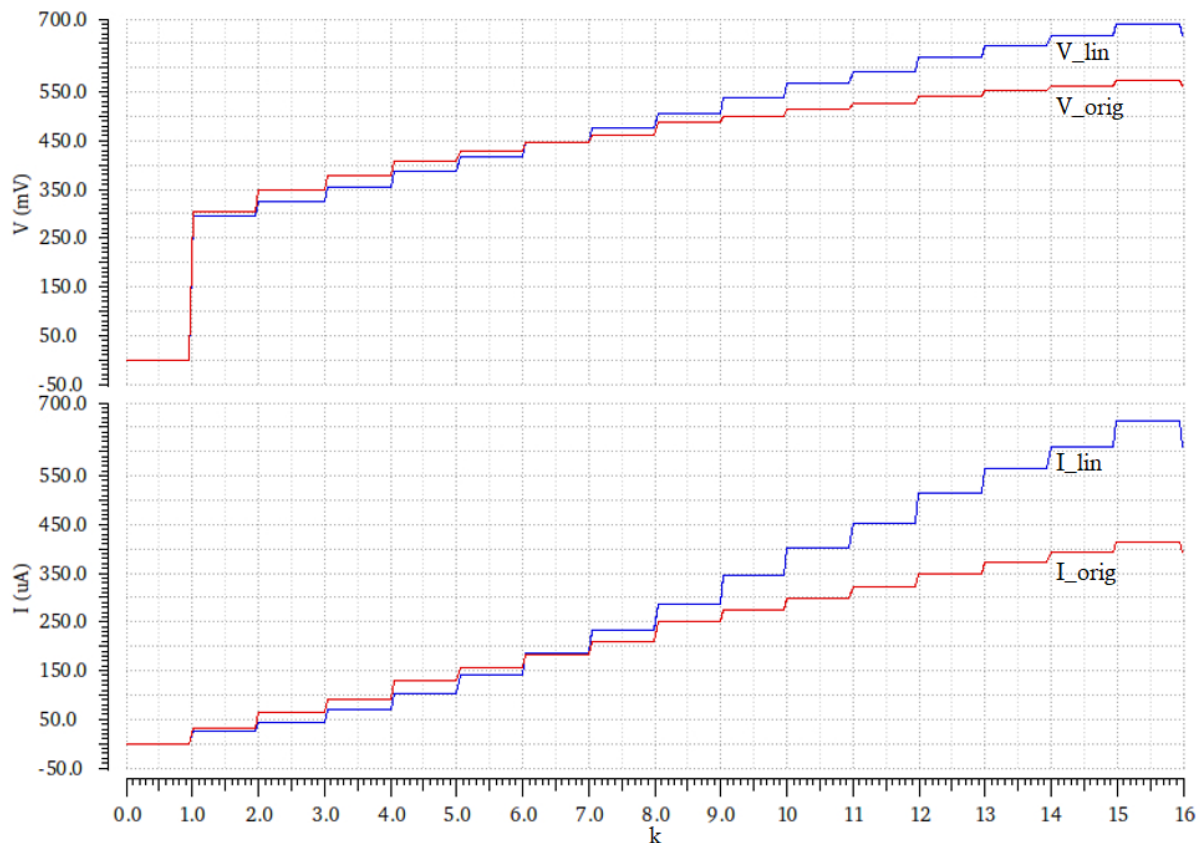
$$I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{tn})^2, \quad (20)$$

missä I_D on transistorin nieluvirta saturaatiossa, μ_n on NMOS transistorin varausten liikkuvuus, C_{ox} on hilaoksidin kapasitanssi, W/L on leveys-pituussuhde, V_{GS} on hila-lähde jännite ja V_{tn} on NMOS transistorin kynnysjännite.

Kuvassa 37 on vielä esitetty vertailu, jossa lähtövirta ja lähtöjännite on piirretty alkuperäisen kanssa samaan kuvaan. Kuvassa näkyy hyvin selkeästi linearisoitu lähtöjännite (musta) ja miten paljon enemmän virtaa piiri syöttää kuin alkuperäinen. Molemmissa simuloinneissa on käytetty samaa mitoitusta diodikytketylle transistorille.



Kuva 36. Virta-DAC lähtöjännitteen linearisoinnin simulointitulokset.



Kuva 37. Virta-DAC vasteen vertailua. Linearisoitu vs. alkuperäinen (punainen = alkuperäinen, sininen = linearisoitu).

4.7 Logiikan ominaisuuksien simulointia

4.7.1 Virta ja tehonkulutuksen mittaus

Komponenttien toiminnan lisäksi on yhtä tärkeää tietää miten kuorma ja ympäristö vaikuttavat toimintaan. Taulukossa 5 on esitetty komponenttien käyttämä staattinen ja dynaaminen maksimivirta ilman kuormaa ja kuorman kanssa. Ilman kuormaa tarkoittaa, että lähtösolmun kapasitanssi on käytännössä komponentin oma lähtökapasitanssi ja näin ollen vaihtelee komponenttien välillä. Kuormana on käytetty 5 kpl minimikokoista invertteriä. Näistä kuorman kanssa mitatuista virroista on laskettu staattinen ja dynaaminen tehonkulutus, joista dynaaminen on mitattu simuloinnissa käytetyllä taajuudella. Kaikkien komponenttien kesken, lukuun ottamatta virta-DAC:ia, on laskettu virtojen ja tehonkulutusten keskiarvot taulukon alimmalle riville. Taulukosta nähdään, että komponentin ottama virta riippuu siitä, montako transistoria siinä kytkeytyy päälle signaalien tilan vaihtuessa. Esimerkiksi schmitt-triggerin kytkeytyessä päälle vain 3kpl transistoreja ottaa hetkellisesti käyttöjännitteestä virtaa, kun taas Inv_X12 tapauksessa transistoreja kytkeytyy yhtä aikaa 12kpl ja hetkellinen virta on paljon isompi.

Taulukon perusteella kuormitettujen komponenttien keskimääräinen staattinen virrankulutus (vuotovirrat) on alle 80 nA ja dynaaminen hieman alle 200 μ A. Maksimissaan virrankulutus on 480 μ A, mikä on vielä reilusti alle virrankestorajojen. Keskimäärin kuormitettujen piirien staattinen tehonkulutus on hieman yli 60 nW ja dynaaminen tehonkulutus noin 2 μ W/MHz. Todellinen tehonkulutus pitäisi laskea kaikkien signaalimuutosten aikana kulutetun virran keskiarvona pidemmältä aikaväliltä ja se kertoisi tarkemmin komponentin

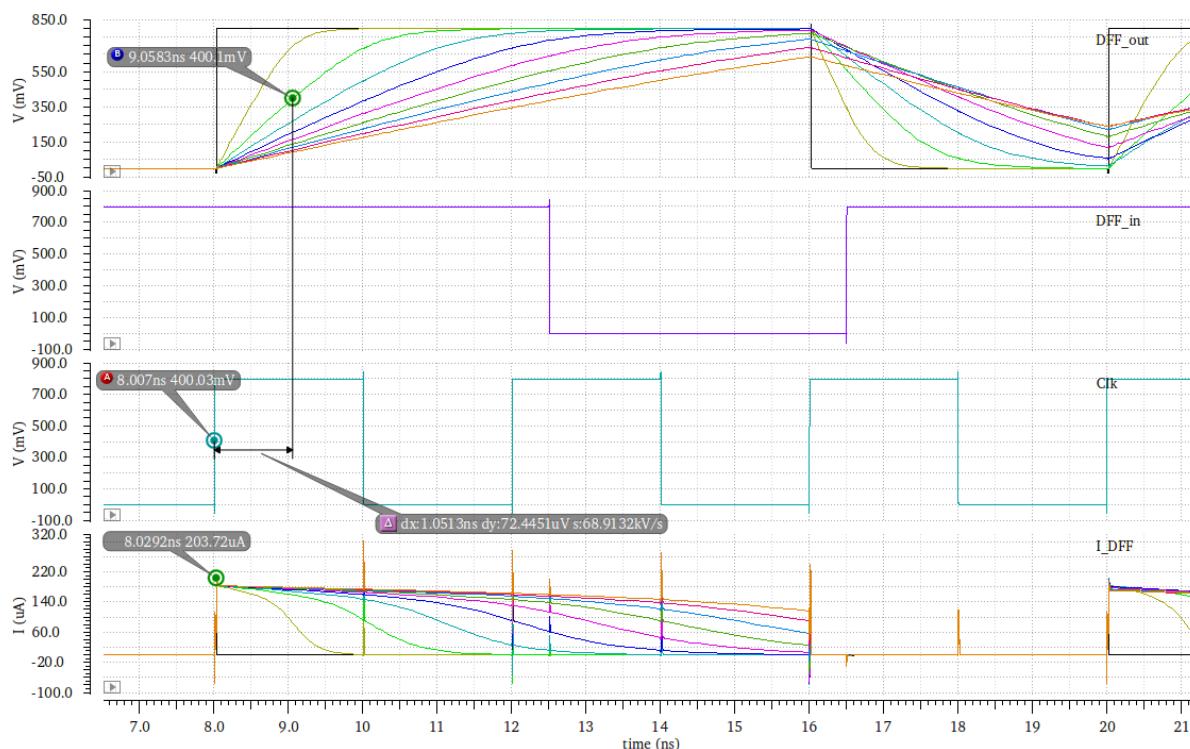
luonteesta. Dynaaminen tehonkulutus kuitenkin kertoo sen, että kytkentätaajuuden kasvaessa kulutettu energia kasvaa huomattavasti. Simuloinnista selviää myös, että ko. prosessin SLVT-transistorit eivät simulointien perusteella kuluta kovin paljon tehoa valmiustilassa.

Taulukko 5. Komponenttien maksimivirrankulutus ja tehonkulutus ilman kuormaa ja kuorman kanssa simuloituna. Taajuudet vaihtelevat simuloitavan lohkon mukaan.

Komponentti	Taajuus [MHz]	Virta ilman kuormaa		Virta 5 inv kuormalla		Tehonkulutus kuorman kanssa	
		Staattinen [nA]	Dynaaminen [uA]	Staattinen [nA]	Dynaaminen [uA]	Staattinen [uW]	Dynaaminen [uW/MHz]
Nand_X2	125,00	27,1	126,8	27,1	172,1	0,0217	1,1014
Nand_X3	62,50	28,9	129,6	28,9	149,2	0,0231	1,9098
Nor_X2	125,00	25,6	99,7	23,8	170,4	0,0190	1,0906
Nor_X3	62,50	26,7	93,2	22,8	166,1	0,0182	2,1261
Mux_2_1	111,00	38,7	152,3	39,2	153,9	0,0314	1,1092
Mux_4_1_BIN/LIN	33,00	130,0	249,4	125,0	249,4	0,1000	6,0461
Schmitt_trigger	125,00	22,3	44,1	29,3	160,9	0,0234	1,0298
DFF	62,50	73,6	121,8	75,1	131,8	0,0601	1,6870
DLatch	150,00	38,9	121,6	38,4	147,4	0,0307	0,7861
And_X2	375,00	14,9	87,8	192,0	107,7	0,1536	0,2298
And_X3	188,00	16,7	117,2	192,0	114,7	0,1536	0,4881
And_X5_LE	31,25	92,9	165,0	270,0	275,9	0,2160	7,0630
Inv_X3	165,00	12,2	75,6	12,2	142,6	0,0098	0,6914
Inv_X9	165,00	41,2	222,4	41,2	373,9	0,0330	1,8128
Inv_X12	165,00	57,0	294,2	57,0	481,4	0,0456	2,3341
CDAC	-	Ei toimi	Ei toimi	57617,0	471,6	46,0936	377,2800
KESKIARVO	-	43,1133	140,0467	78,2667	199,8267	0,0626	1,9670

4.7.2 Kuorman vaikutuksia toimintaan

Kuvassa 38 nähdään D-kiikun lähtöjännitteet ja kuluttama virta erilaisilla kapasitiivisilla kuormilla. Kuormakapasitanssin arvo on simuloinnissa pyyhkäisty välillä 1 fF – 2 pF käyttäen 10 välipistettä. Kuvan perusteella voidaan nähdä helposti, että kapasitiivisen kuorman kasvaessa lähtösignaali hidastuu merkittävästi ja liian suuren kapasitiivisen kuormituksen edessä lähtösignaali ei ehdi nousta ollenkaan ylös. Liian hidas signaali leikkautuu. Kuvasta nähdään myös selkeästi se, että komponentti kuluttaa käyttöjännitteestä virtaa vain kuormakapasitanssia tai piirin sisäisten solmujen kapasitansseja ladataessa. Kuormakapasitanssin kasvaessa lataus on hitaampaa ja virtaa kulutetaan sen aikaa kuin lataus tapahtuu (I_DFF käyrä), mutta varauksen purkaminen ei lähtösolmussa ota virtaa käyttöjännitteestä. Tämä voidaan nähdä virtakäyrästä tilanteessa, jossa signaali DFF_out on matalassa potentiaalissa. Dynaaminen virrankulutus näkyy tilanvaihtojen aiheuttamina virtapiikkeinä ja staattinen virrankulutus on tilanvaihtojen välissä kuluva virta.



Kuva 38. D-Kiikun lähtöjännitteen ja virrankulutuksen simulointitulokset eri kuormakapasitanssin arvoilla.

Kuvassa 38 olevaa vihreää käyrää vastaava kuormakapasitanssin arvo on 445,2 fF ja tällä kuormalla mitattu viive kellosignaalin muutoksesta lähtösignaalin muutokseen on 1,051 ns.

Lohkon teoreettisen viiveen laskemiseen kyseisellä kuormalla voidaan hyödyntää ensimmäisen asteen RC-piirimallia. Kaavasta (13) voidaan laskea aktiiviresistanssi PMOS transistorille hyödyntäen kuvan virta-arvoa.

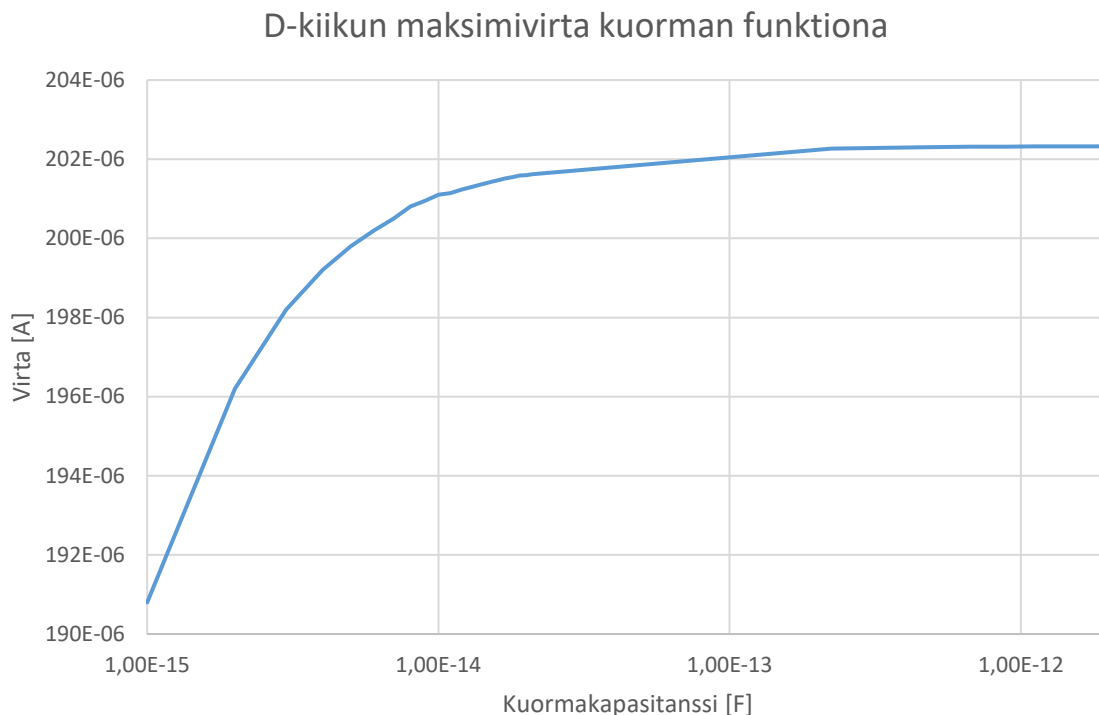
$$R_{on,PMOS} = \frac{800 \text{ mV}}{203,72 \text{ }\mu\text{A}} = 3,927 \text{ k}\Omega, \quad (21)$$

ja kaavasta (12) puolestaan voidaan laskea viive

$$t = \ln 2 \cdot R \cdot C_L = 0,69 \cdot 3,927 \text{ k}\Omega \cdot 4,452 \cdot 10^{-13} \text{ F} = 1,2118 \text{ ns}. \quad (22)$$

Todellinen kuormakapasitanssi ja solmun kokonaisresistanssi poikkeavat hieman laskuissa käytetyistä ja lasketuista arvoista, sillä niissä ei ole huomioitu D-kiikun omaa lähtökapasitanssia, eikä PMOS aktiiviresistanssin rinnankytkentää NMOS kiinni-resistanssin ja johtimista aiheutuvan resistanssin kanssa. Tämän vuoksi kuvassa näkyvä viive 1,051 ns poikkeaa hieman lasketusta 1,212 ns arvosta. Laskettu arvo kuitenkin osoittaa sen, että ensimmäisen asteen RC-piiriä voidaan käyttää logiikkaporttien viiveiden arviointiin.

Kuvassa 39 nähdään D-kiikun virrankulutuksen kasvu kuormakapasitanssin funktiona välillä 1 fF – 2 pF. Kuormakapasitanssin kasvaessa yli 10 fF virrankulutus alkaa tasoittumaan ja kasvu pysähtyy noin 1 pF jälkeen. Tämä johtuu siitä, että kuorman lataaminen käyttää kaiken sen virran, jota transistorit pystyvät antamaan eli käytettyjen transistorien rajat tulevat vastaan, jolloin lähtösolmun jännite nousee hitaasti ylös. Mitatulla aikavälillä muutos on lähes lineaarinen isoilla kuormakapasitanssin arvoilla.

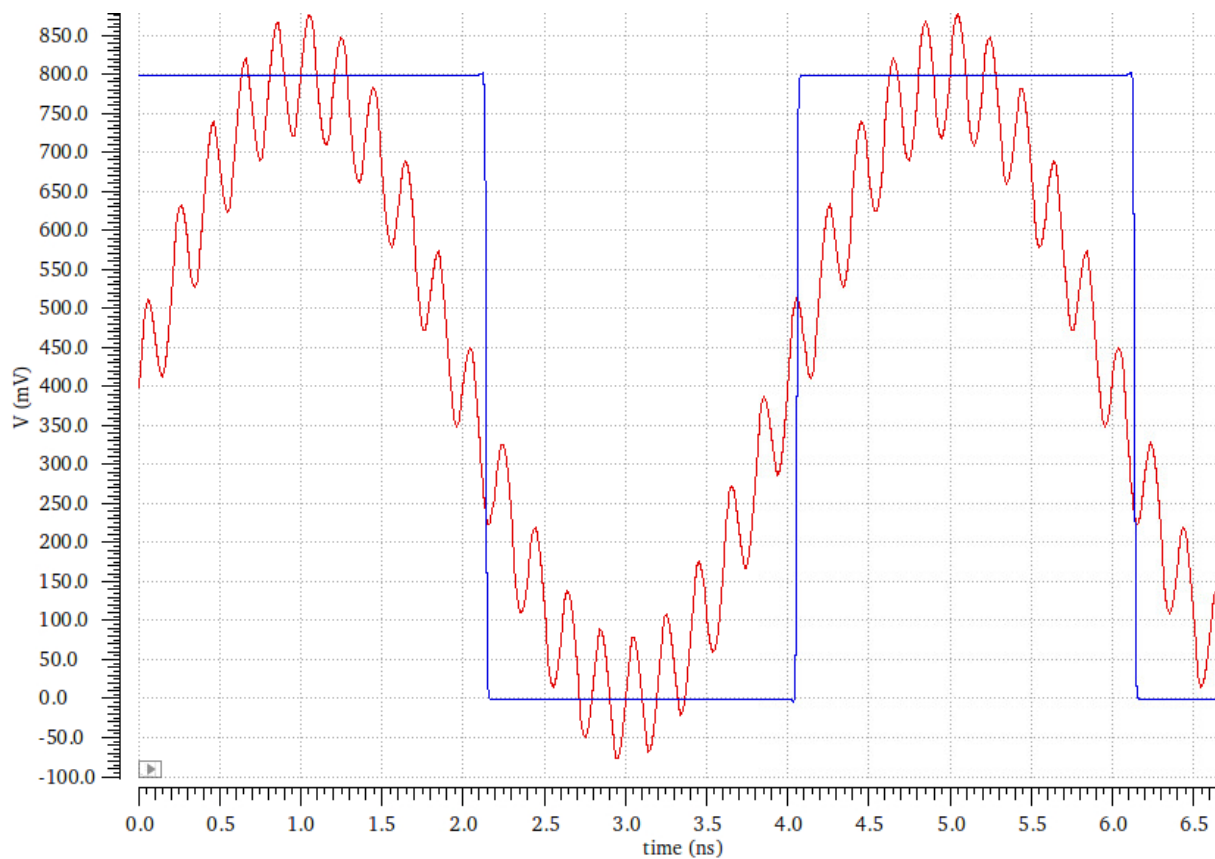


Kuva 39. D-Kiikun maksimivirta kuormakapasitanssin funktiona.

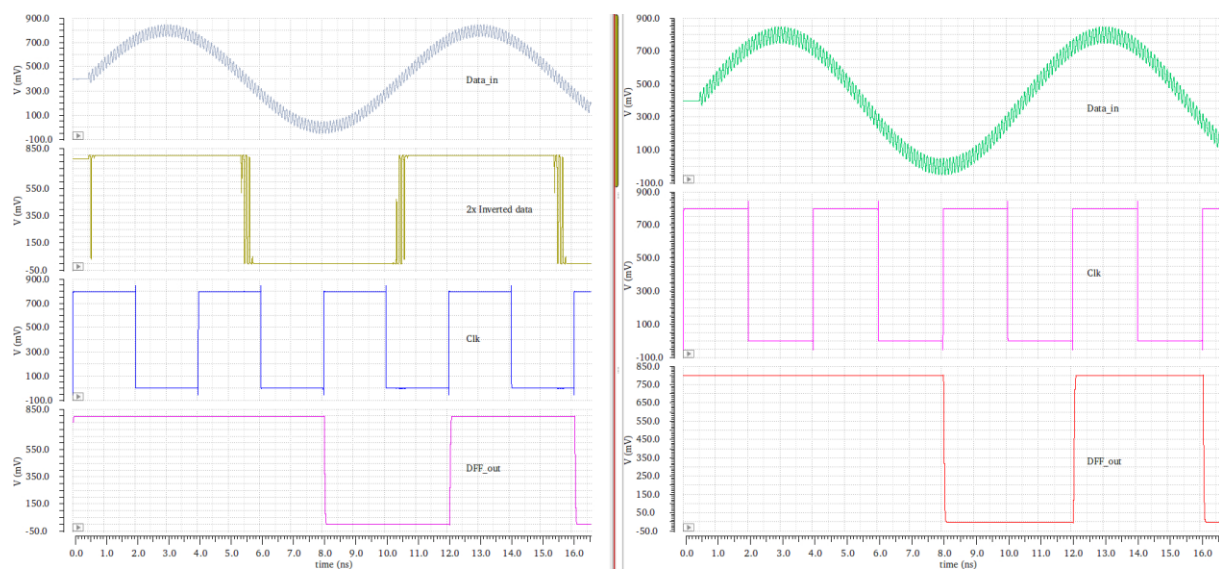
4.7.3 Kohinan vaikutuksia toimintaan

Kuvissa 40 ja 41 on havainnollistettu logiikan kykyä suodattaa kohinaa. Kuvan 40 simuloinnissa schmitt-triggerin tuloon ajetaan hidasta signaalia, johon on sisällytetty AC-muotoista kohinaa. Schmitt-triggerin lähtö vaihtaa tilaansa tulosignaalin ylittäessä kynnysjännitteen arvon ja kohinasignaali ei aiheuta kynnysjännitteen ylittymistä uudelleen. Simuloinnissa kohinan amplitudi on 80 mV ja schmitt-triggerin kynnysjännitteiden välinen ero on n. 90 mV eli komponentti toimii juuri niin kuin pitääkin. Kuvan 41 vasemmanpuoleisessa simuloinnissa on ajettu AC-signaalia, jossa korkeampitaajuinen AC-signaali mukana, D-kiikun tuloporttiin kahden invertterin läpi ja oikealla suoraan tuloporttiin. Vasemmanpuoleisesta kuvasta näemme, että invertteri itsessään jo muokkaa tulosignaalia enemmän logiikkamaiseksi, mutta kohinan amplitudin aiheuttamana sekä nousevalla, että laskevalla reunalla näkyy useita tilanvaihdoksia. D-kiikun lähtö kuitenkin pysyy siistinä, sillä kiikku lukee tulosignaalin arvon kellon nousevalla reunalla ja ei välitä muuna aikana tapahtuvista asioista. Samankaltainen luonne on lähes kaikilla logiikkakomponenteilla. Mitä lähempänä datasignaalin reunaa kellon nouseva reuna on, sitä suurempi todennäköisyys on virheelliselle tulokinnalle useitten tilanvaihdosten takia.

Kuvista nähdään myös, että tulosignaaleissa olevat pienet jännitemuutokset eivät siirry lähtöön, koska logiikan satureoituva vahvistus vaimentaa kohinaa. Joskus tulo- ja lähtösolmuihin lisätään suunnitteluvaiheessa kondensaattori, joka entisestään hidastaa ja vaimentaa signaaleja, jolloin äkilliset virta- ja jännitemuutokset eivät näy komponentin toiminnassa. Tästä hyvä esimerkki on virtapeili, jossa tulovirta halutaan mahdollisimman tarkasti kopioida lähtöön tietyllä peilaussuhteella. Jos ajetaan virtaa johonkin kuormaan ja samaan aikaan piirillä iso määrä muuta logiikkaa vaihtaa tilaansa, voi virrassa näkyä hyvinkin isoja hetkellisiä piikkejä. Nämä piikit puolestaan voivat hyvinkin herkästi heilauttaa jotain jännitettä logiikkatason väärälle puolelle tai särkeä johtimia, jos niitä ei suodateta pois.



Kuva 40. Tulosignaalin kohinan vaikutus Schmitt-triggerin toimintaan.



Kuva 41. Tulosignaalin kohinan vaikutus D-kiikun toimintaan.

5 PIIRIKUVIOIDEN SUUNNITTELU

Kun tarvittavat lohkot oli suunniteltu ja simuloitu, aloitettiin piirikuvioiden suunnittelu. Tavoitteena oli luoda sellaiset rakenteet, että muilla käyttäjillä on mahdollisimman helppo lisätä niitä omiin suunnitelmiinsa ja jotka ovat mahdollisimman yksinkertaista kytkeä osaksi isompaa piirilohkoa. Rakenteet suunniteltiin siten, että kaikkien komponenttien korkeus ja leveys noudattavat kolmannen metallikerroksen jakoa. Komponenttien korkeus määritettiin monimutkaisimman rakenteen perusteella.

Yleisellä tasolla kaikkien rakenteiden suunnittelussa noudatettiin samoja periaatteita ja piirikuvioiden suunnittelussa piti ottaa huomioon myös muiden rakenteiden ominaisuudet. Tämä tarkoittaa sitä, että esimerkiksi monimutkaisemmissa rakenteissa kytkentöihin vaadittava korkeus määrää myös yksinkertaisten piirirakenteiden korkeusmitat, jolloin suunnittelua aloittaessa pitää rakenteeseen jättää varaa muokata esimerkiksi käyttöjännitekiskojen paikkaa.

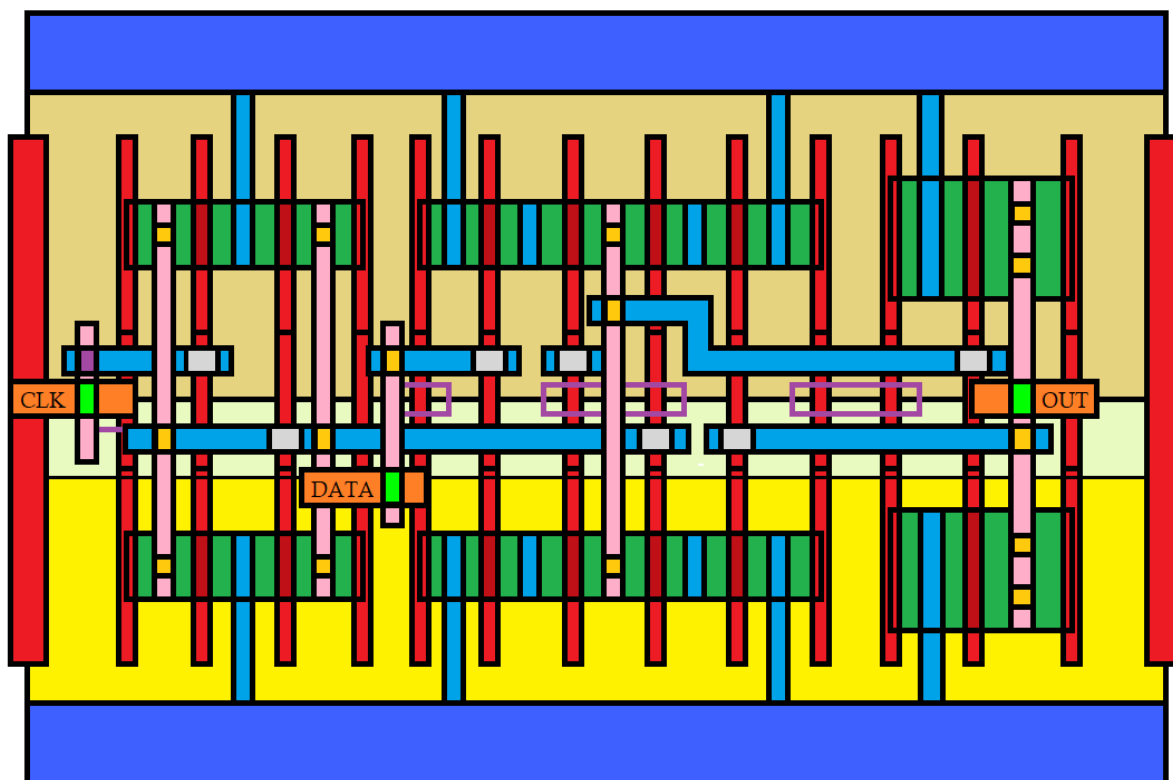
5.1 Suunnittelun periaatteet ja tavoitteet

Suunnittelun tavoitteena oli rakentaa mahdollisimman pieniä ja tiiviitä piirikuviota, jotka toimisivat mahdollisimman nopeasti. Useimmat valmiit transistorimallit olivat pinta-alaltaan niin suuria, ettei niitä kannattanut käyttää. Valmiissa malleissa oli korkeille metallikerroksille nostetut pinnit ja niiden ympärillä oli suojarenkaita, joilla varmistettiin transistorin rungon substraattikontakti ja toiminta. Valmiiden transistori piirikuvioiden rakenteiden suuren koon vuoksi päädyttiin poimimaan niistä vain olennaisin osa, substraattikontaktiin tarvittavat kerrokset, muuten käytettiin pienimpiä transistorin piirikuviomalleja ilman suojarenkaita. Substraattikontaktille rakennettiin oma erillinen piirikuvio, joka lisättiin jokaiseen rakenteeseen luomaan tarvittava kontakti, ja suojarenkaat lisättiin vasta varsinaisiin piirikuvioiden RFIC-komponenttien kanssa. Pienen koon tavoittelu johti myös siihen, että kaikki piirikuviot tehtiin kustomoituna, eli esimerkiksi kiikun rakentamisessa ei käytetty salvan piirikuviota apuna, vaan kaikki transistorit yhdisteltiin käsin erikseen. Näin saatiin pudotettua pinta-alaa yli 50% pienemmäksi.

Kuvassa 42 on hahmotelma D-salvan piirirakenteesta. Vihreä alue kuvastaa transistorin aktiivialuetta, punaiset johtimet transistorin yli ovat hilamateriaalia eli polyä. Nielu ja lähde vuorottelevat hilakontaktien molemmiin puoliin. Hilan leveys ja pituus määräytyvät vihreän alueen päällä olevan hilan osan pituudesta ja leveydestä (tummempi punainen). Kuvasta nähdään, että transistorien päissä oleva poly-materiaali on asetettu siten, että se on puoleksi transistorialueen päällä ja puoleksi substraatin päällä, tämä rakenne katkaisee transistorin aktiivialueen ja on suunnitteluohjelmissa yleensä selkeästi merkitty käyttäen omaa polykerrosta. Jos erityyppisten transistorien päissä olevat nielut tai lähteet on kytketty eri potentiaaleihin tai transistorien hilat on kytketty eri johtimiin, täytyy kontaktit katkaista N- ja P-alueiden välissä käyttämällä katkaisua tarkoittavaa kerrosta CT (cut). Tämä on esitetty kuvassa violetilla suorakulmiolla tilanteessa, jossa transistorien viimeiset lähteet on kytketty eri potentiaaleihin. Koko rakenteen molempiin laitoihin on laitettu vielä dummy-polyt, jotka tarvitaan estämään muiden rakenteiden tuonti liian lähelle transistorin aktiivialuetta ja näin vähentämään esimerkiksi ympäristön vaikutusta transistorin toimintaan. Transistorien sijoittelu tapahtuu siten, että PMOS-transistorit ovat käyttöjännitekiskon (leveä sininen johdin ylhäällä) alapuolella vaalean ruskealla merkityn p-substraatin päällä ja NMOS-transistorit ovat maakiskon (leveä sininen johdin alhaalla) yläpuolella keltaisella merkityn n-altaan päällä.

Varsinaiset johdotukset tapahtuvat siten, että esimerkiksi hilalta nostetaan johdin läpiviennin avulla (harmaa neliö sinisessä johtimessa) ensimmäiselle metallikerrokselle ja metallijohtimen

avulla hila yhdistetään sitten toisen transistorin nieluun kiinni käyttäen kahta alinta metallikerrosta. Johdotukset pyrittiin tekemään siten, että ensimmäisellä metallikerroksella johtimet piirretään vaakatasossa ja toisella metallikerroksella pystysuunnassa. Hilojen välisiä johdotuksia tehtiin käyttämällä polyä johtimena, mutta hilalta nielulle tai lähteelle piti aina tehdä johdotus metallijohtimella. Kuvassa on havainnollistettu esimerkkinä johtimia, jotka on yhdistetty eri paikkoihin käyttämällä edellä mainittua periaatetta. Tällä tavalla rakentaessa voidaan hyödyntää tehokkaasti kunkin metallikerroksen johtimien välistä minimietäisyyttä. Johtimien risteyskohdassa olevat oranssit pisteet ovat läpivientejä ensimmäiseltä metallikerrokselta toiselle. Tuloportit (oranssi leveä johdin) nostettiin kolmannelle metallikerrokselle siten, että niiden kytkeminen on mahdollisimman helppoa.



Kuva 42. D-Salvan piirikuvion hahmotelma.

5.1.1 Piirikuvioiden koko

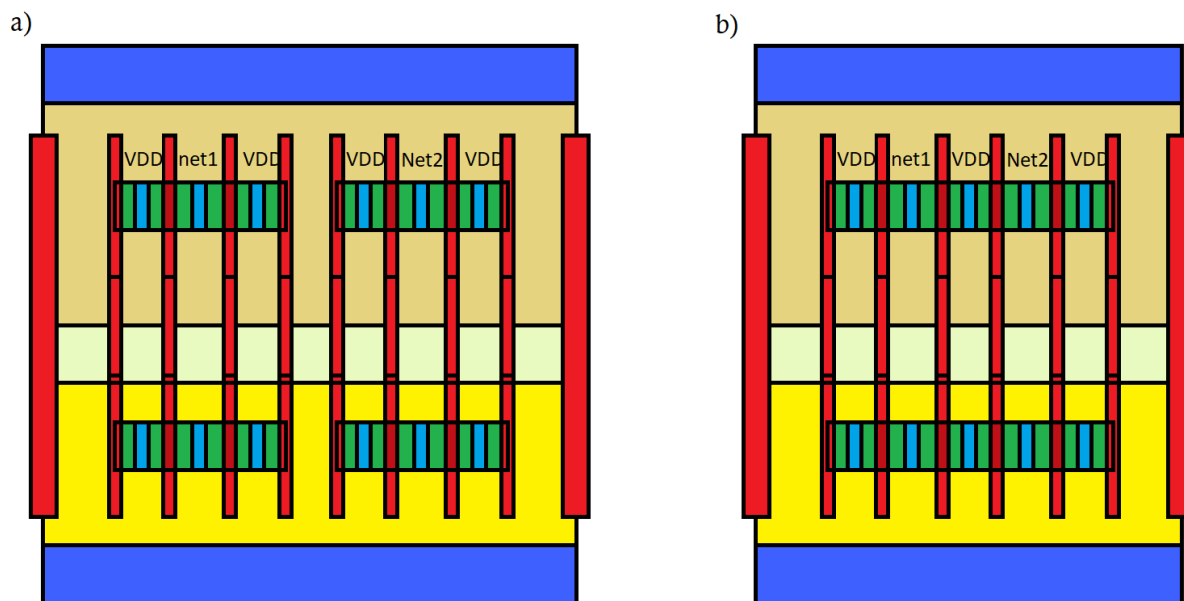
Piirikuvioiden koko on digitaalilogiikassa merkittävä tekijä. Tässä työssä piirikuvioiden koko minimoitiin tekemällä paikalliset johdotukset mahdollisimman matalalla metallikerroksella. Myös hilamateriaalia, polyä, käytettiin apuna. Transistoreja sijoiteltaessa paikoilleen niitä pyrittiin myös yhdistelemään mahdollisimman paljon, eli niiden yhteisiä solmupisteitä jaettiin transistorien kesken. N- ja P-tyypin transistorit tuotiin niin lähelle vastakkaista aluetta kuin mahdollista. Piirien koon haluttiin noudattavan kolmannen metallikerroksen minimijakoa, jolloin monimutkaisinta piirikuviota kasvatettiin pystysuunnassa sen verran, että korkeus istui haluttuun jakoon. Lopullinen korkeus olikin vain $1.71\ \mu\text{m}$ ja kaikki komponentit olivat sen korkuisia. Leveys puolestaan oli suoraan verrannollinen laitteen kompleksisuuteen, mitä enemmän transistoreja, sitä leveämpi rakenne.

Niissä komponenteissa, joissa tarvittiin ulkoista suodatusta tai joiden rakenteet sisälsivät useita eri komponenttien piirikuviota, kuten esimerkiksi siirtorekisteri, rakenne suunniteltiin siten, että se on mahdollisimman järkevä ja käytännöllinen. Esimerkiksi siirtorekisterissä komponentit pinottiin kahteen kerrokseen, jolloin leveys saatiin kutistettua puoleen ja kyseistä komponenttia on myös helpompi käyttää. Virta-DAC:n rakenteesta tehtiin neliön mallinen siten, että komponentin kokonaiskorkeus on n -kertainen $1.71\ \mu\text{m}$ korkeuteen nähden ja leveys määräytyi transistorien mukaan.

Nopeus on toinen asia logiikassa, joka täytyy huomioida. Tässä työssä käytettiin minimikokoisia SLVT-tyypin transistoreja, joiden kynnysjännite on matala ja ne ovat nopeita. Piirikuvioiden nopeuteen pystyttiin vaikuttamaan paljon sillä, että käytettiin pääasiassa minimilevyisiä johtimia ja mahdollisimman lyhyitä reittejä. Tämä vaikutti viiveeseen ja nopeuteen RC-aikavakion osoittamalla tavalla. Usein suunnitelmissa jouduttiin valitsemaan pinta-alan ja nopeuden väliltä. Esimerkiksi 4-tuloinen multiplekseri rakennettiin käyttäen BIN/LIN-koodausta, koska sillä saatiin nopeutta kasvatettua 30%. Koodauslohko kuitenkin lisäsi transistorien määrää huomattavasti ja kasvatti pinta-alaa.

5.2 Rakenteiden tekeminen

Jokaisen komponentin piirikuvioiden rakentaminen noudatti samaa ajatuskulkua. Ensin yhdisteltiin transistoreja mahdollisuuksien mukaan siten, että jaettiin tehokkaasti yhteisiä lähdekontakteja. Esimerkiksi, jos kahdessa samankokoisessa transistorissa on molemmissa lähde kytketty käyttöjännitteeseen, voidaan kyseiset transistorit yhdistää siten, että ne käyttävät samaa lähdetä, mutta hilat lähtevät eri suuntiin. Kuvassa 43 on havainnollistettu, miten rakenteita voi yhdistää. Yhdisteleminen vaati sen, että transistorit ovat saman levyisiä, eli niiden aktiivialue on saman korkuinen. Kuvan a) ja b) rakenteet ovat toiminnaltaan identtiset, mutta kuvassa 43 b) on jaettu yksi lähdekontakti. Mitoitusvaiheessa työssä huomioitiin piirikuvioiden tekeminen siten, että isoimmissa transistoreissa kasvatettiin sormien lukumäärää niin paljon, että päästiin lähelle pienien transistorien kokoluokkaa. Kaikki suuret transistorit mitoitettiin lopulta siten, että yhden sormen maksimileveydeksi tuli $240\ \text{nm}$, mikä mahdollisti kokonaiskorkeuden asettamisen tiettyyn arvoon jo ennen kuin muut rakenteet olivat valmiita.



Kuva 43. Transistorien yhdistämisen havainnollistaminen, a) transistorit erillään, b) transistorit jakavat yhden VDD kontaktin. Kummassakin rakenteessa on kaksi invertterää dummy-hiloineen.

Kun transistorirakenteet oli yhdistelty ja sijoiteltu siten, että PMOS ja NMOS transistorit ovat omissa lohkoissaan, kytkettiin niiden hilat yhteen polyllä. Tärkeää tässä vaiheessa oli se, että yhdistelemällä tuotetuissa rakenteissa oli yhtä monta hilaa/sormea molemman tyyppisissä transistoreissa. Mikäli p- ja n-tyypin hilojen määrä poikkesi toisistaan, luotiin piirille dummy-transistoreja tai dummy-polyjä, joilla kompensoitiin poikkeama. Rakenteiden täytyi myös olla ympäristöltään mahdollisimman symmetriset, jotta jokaisen transistorin toiminta olisi samanlaista ja välttyttäisiin DRC-virheiltä (Design Rules Check). Seuraava vaihe oli nostaa kaikki hilakontaktit ensimmäiselle metallikerrokselle ja sijoitella läpiviennit siten, että saadaan kaikki läpiviennit mahtumaan PMOS- ja NMOS-transistorien väliin jäävään tilaan. Tässä vaiheessa myös katkaistiin sellaiset hilakontaktit, jotka eivät saa olla oikosulussa keskenään.

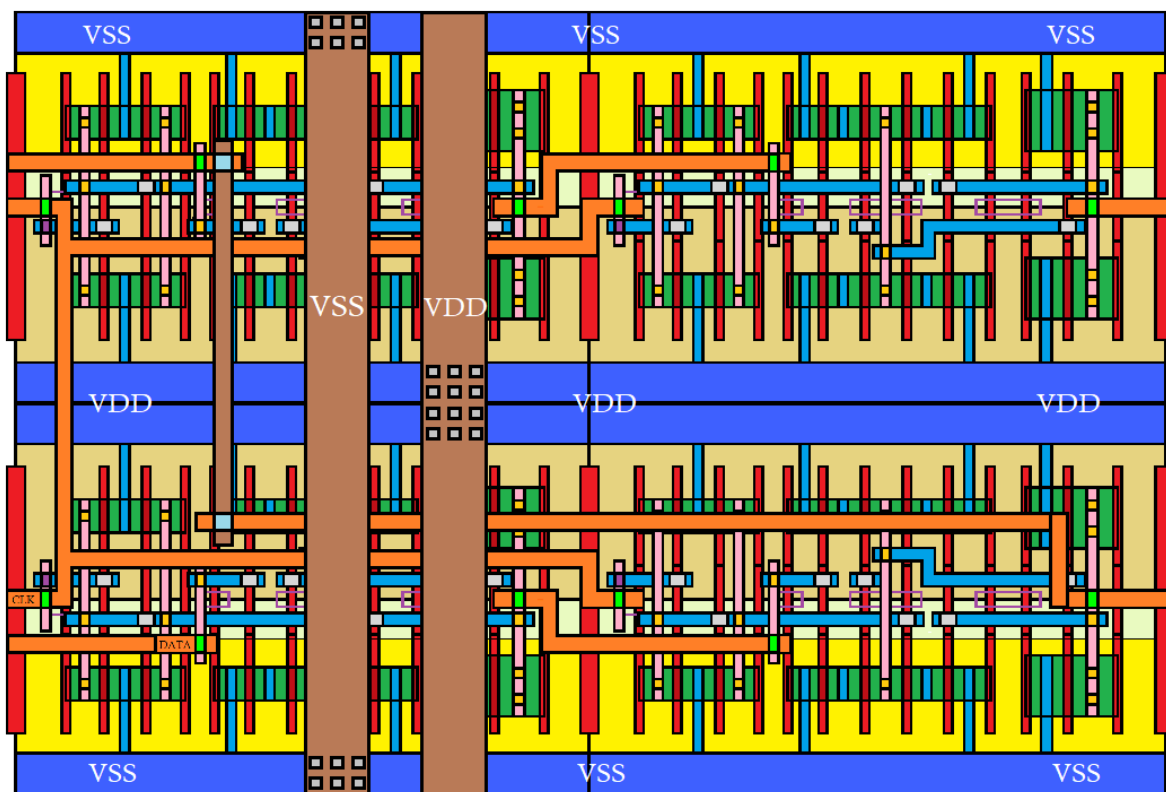
Johdotuksessa täytyi osata ottaa monia asioita yhtä aikaa huomioon. Yksi vaikeimmista asioista oli se, että koko ajan piti hahmottaa mikä johdin kuuluu mihinkin ja kuinka moneen paikkaan se pitää kytkeä. Toinen asia oli se, että täytyi pystyä näkemään se, mistä johdoista kytkennät aloitetaan, koska sillä oli suuri vaikutus lopulliseen rakenteeseen. Tässä työssä usein mietittiin ensin, mitkä johtimet tulevat olemaan pisimpiä ja tehtiin niille reitti, sen jälkeen tehtiin lyhyimmät johdotukset mahdollisimman vähillä läpivienneillä, eli toisin sanoen käytettiin pääasiassa ensimmäistä metallikerrosta. Tämän jälkeen yhdisteltiin loput johtimet siten, että ne kulkevat mahdollisimman lähellä toisiaan vaaka- ja pystysuunnassa. Piirirakenteiden johdotusten tekeminen oli melko vapaata. Niissä rakenteissa, joissa oli vähän kontakteja, pystyi käyttämään tilaa enemmän ja monimutkaisissa rakenteissa jokainen johto piti miettiä erikseen hyödyntäen metallikerrosten minimietäisyyksiä.

Kolmas asia johdotuksessa oli se, että koko ajan täytyi pitää mielessä kuhunkin kerrokseen liittyvät säännöt. Säännöt liittyivät esimerkiksi johtimien leveyksiin ja niiden välisiin etäisyyksiin. Myös käytetyn teknologian tuomat virta- ja jänniterajoitukset täytyi huomioida samoin kuin mahdolliset tehon siirtoon ja lämpenemiseen liittyvät asiat. Näitä on esimerkiksi käyttöjännitekiskojen suunnitteleminen siten, että ne ovat tarpeeksi leveitä antamaan koko komponentille riittävästi virtaa. Suurissa rakenteissa, joissa oli useita komponentteja ladottuna

vierekkäin ja useita eri käyttöjännitekiskoja, täytyi myös kiskot yhdistää sopivan levyisillä johtimilla, jotta tehonjakoverkko toimisi mahdollisimman hyvin.

Kokonaiskuvan hahmottaminen oli oma osansa suunnittelua. Esimerkiksi kiikun ja salvan suunnittelussa täytyi pinnien sijoittelu tehdä siten, että siirtorekisteriä rakentaessa voi useita tällaisia komponentteja kytkeä helposti peräkkäin. Suunnittelussa päädyttiinkin ratkaisuun, jossa rakenteiden kellosignaalit pidetään tietyllä korkeudella, jolloin kellojohdin voidaan isoissa rakenteissa vetää suoraan koko komponenttikokoelman yli ilman turhia mutkitteluja. Vastaavanlaista sijoittelua tehtiin myös lähtösignaalien ja tulosignaalien osalta kiikussa, jossa lähtö ja tulo sijoitettiin samalle tasolle, jotta ne voidaan kytkeä suoralla johtimella toisiinsa.

Kuvassa 44 on esimerkkiahmotelma kahden kiikun päällekkäin sijoittamisesta ja käyttöjännitekiskojen yhdistämisestä. Kuvasta nähdään, että käyttöjännitteet tuodaan leveällä johtimella piirin päälle, josta sitten haaroitetaan jokaiseen komponenttiriviin ja sieltä edelleen tarvittaviin kytköksiin transistoreille. Näin saadaan jännite pysymään tasaisempana koko piirirakenteen yli. Kuvasta nähdään, että käyttöjännite- ja maakisko vuorottelevat, joten jokaisen rivin komponentit täytyy olla käännettynä ympäri suhteessa seuraavaan. Tämä mahdollistaa sen, että saadaan tehtyä p- ja n-tyypin altaat omiin isompiin kokonaisuuksiin ja näin rakenne pysyy järkevämpänä. Isoissa rakenteissa, kuten siirtorekisterissä ja osoitekooderissa täytyi johdotuksia tehdä käyttöjännitekiskojen yli eri komponenttien välillä.



Kuva 44. Kahden päällekkäin asetetun kiikun piirikuvion hahmotelma.

5.3 Calibre nmLVS tarkistus

LVS (Layout Versus Schematic) -tarkistuksessa käydään piirikuvion johtimet ja rakenteet läpi ja niistä muodostetaan tietorakenne (netlista), jota sitten verrataan piirikaaviosta ekstraktoituun tietorakenteeseen. Tämä on tehokas tapa selvittää, onko piirissä esimerkiksi kytkemättömiä johtimia, onko johtimia oikosulussa keskenään tai puuttuuko joitain johtimia.

Tarkistus tapahtuu siten, että ensin ohjelma lukee piirikuvion tietokantatiedostot, lukee sieltä kaikki metallikerrokset, käytetyt komponentit ja pinta-alat, joista sitten generoi tiedot siitä, miten kytkennät on tehty, mitä komponentteja on käytetty ja onko kytkennöissä ristiriitoja. Kun tietokannasta on muodostettu piirin komponentit ja johtimet käsittävä tiedosto, voidaan sitä verrata piirikaavion vastaavaan tiedostoon. Näin huomataan heti, onko jonkun komponentin ominaisuudet erilaiset kuin piirikaaviossa, puuttuuko komponentteja, onko niitä liikaa, onko johtimia oikosulussa keskenään ja onko piirillä kytkemättömiä tai avoimia johtimia. Näistä ohjelma luo raporttiin kaikki havaitut puutteet ja virheet sekä mahdollistaa myös sen, että kyseiset virheet pystytään käyttöliittymän avulla paikantamaan piiriltä esimerkiksi korostamalla tietyn johtimen alue eri värillä. Jos tarkistuksessa ei löydy epäkohtia, ohjelma palauttaa ilmoituksen, että kaikki on kunnossa.

5.4 Calibre nmDRC tarkistus

DRC-tarkistus on piirin valmistajan luoma sääntökokoelma, joka kattaa hyvin monipuolisesti käytetyn teknologian eri kerroksiin liittyviä ominaisuuksia, joiden puitteissa piirikuvion tulee pysyä. Suunnitellessaan valmistusprosessia ovat sen kehittäjät testanneet käytännössä kaikki mahdolliset kombinaatiot mitkä valmistuksessa toimivat ja mitkä eivät toimi. Esimerkkinä eri materiaalien minimietäisyydet ja mahdolliset sijoittelumahdollisuudet, sekä miten kerroksia voi laittaa päällekkäin tai limittäin. Testattuihin ominaisuuksiin perustuen on luotu säännöt, jotka täytettyään pitäisi valmistettavan piirin valmistusprosessin näkökulmasta toimia. Tämä valmistusprosessiin liittyvä säännöstö voidaan jakaa kolmeen alueeseen, kerrosten sisäiset säännöt (Intralayer Constraints), kerrosten väliset säännöt (Interlayer Constraints) ja erityissäännöt (Special Constraints).

Tietyn kerroksen sisäisiä DRC-sääntöjä ovat esimerkiksi johtimien minimileveydet, minimivälit, minimipinta-alat ja tuplamaskeihin liittyvät minimisäännöt eli nämä säännöt määrittävät jokaisen kerroksen objektien mitat ja minimivälit eri objektien välillä. Toisen alueen sääntöihin on esimerkiksi eri kerrosten päällekkäisyyksiin, niiden välisiin läpivienteihin, kontakteihin, laajennuksiin ja etäisyyksiin liittyviä asioita. Tämä säännöstö kattaa siis transistorien rakentamiseen liittyvät säännöt, kerrosten toisiinsa kytkemiseen liittyvät säännöt ja allas sekä substraattisäännöt. Kolmanteen osioon voidaan sisällyttää erikoissäännöt, kuten antennisäännöt ja tiheyssäännöt. Näistä esimerkkinä tilanteet, joissa poly-kerroksen läpivienti ensimmäiselle metallikerrokselle tulee tehdä riittävän lähellä transistorin aktiivialuetta, jotta varaukset pääsevät purkautumaan oikein ja se, että eri metallikerroksilla isot tyhjät alueet tulee täyttää metallisilpulla. [3] [17]

Myös DRC-tarkistus ajetaan ohjelmalla läpi ja siinä luetaan piirikuvion tietokannasta rakenteiden ominaisuudet. Luettuja ominaisuuksia verrataan DRC-sääntöihin, jotka on kirjoitettu omiin tietokantoihin ja jos tarkistuksessa ilmenee ongelmia tai virheitä, palauttaa ohjelma niistä raportin. Käyttöliittymä myös auttaa paikantamaan virheet korostamalla ne ja kertoo mitä sääntöä rikottiin sekä myös antaa mahdollisia ohjeita esimerkiksi minimimitoista ja etäisyyksistä liittyen havaittuun virheeseen.

5.5 Parasiittisten ekstraktointi Calibre xACT:n avulla

Kuten jo aiemmin käsiteltiin, voidaan jokainen piiri mallintaa vastusten, kelojen ja kondensaattorien avulla. Calibren xACT-ohjelman avulla voidaan piirikuviosta muodostaa piirikaavio, joka sisältää johtimien ja komponenttien resistiiviset ja kapasitiiviset kuten myös

induktiiviset ominaisuudet. Näiden kokoon vaikuttaa esimerkiksi johtimien leveydet, pinta-alat, käytetyt materiaalit ja monet muut ominaisuudet.

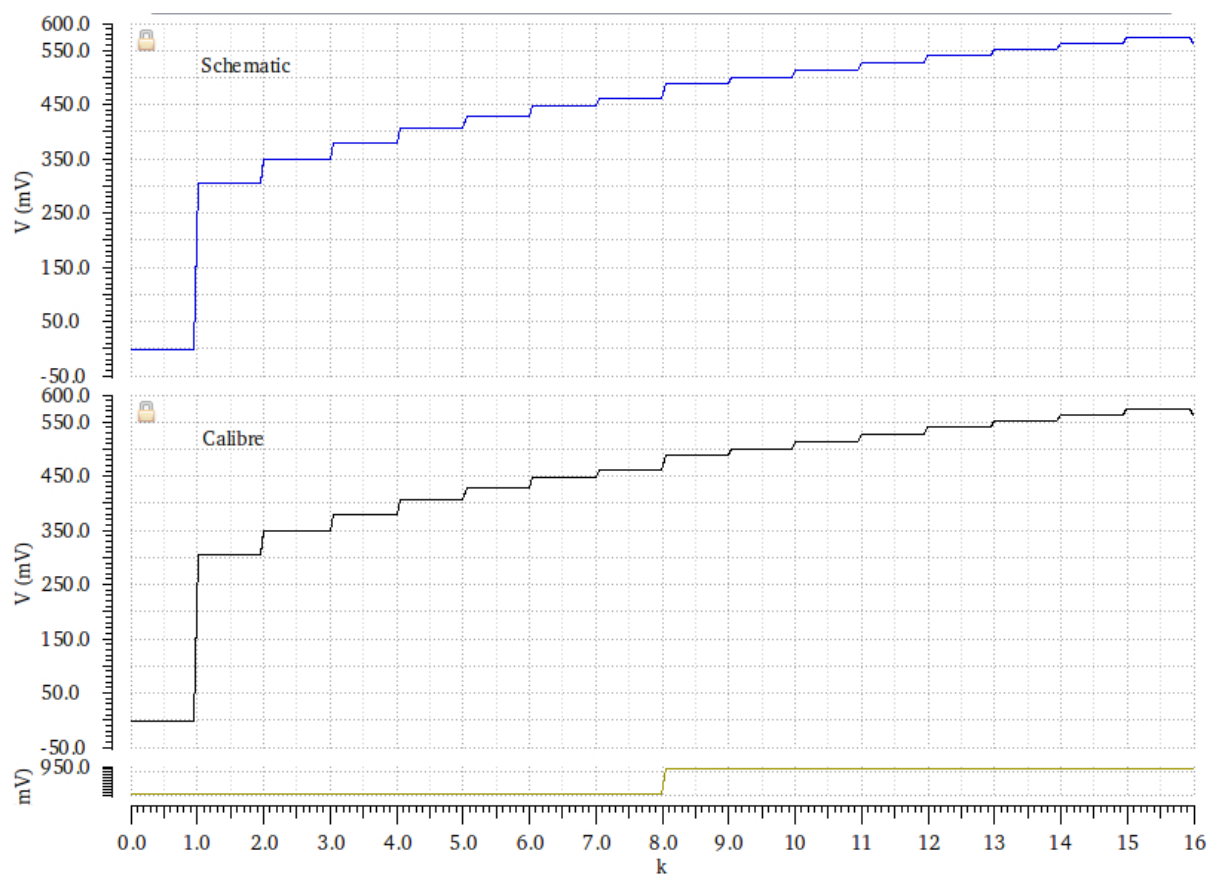
Sen lisäksi, että ohjelma arvioi johtimien ja komponenttien ominaisuuksia, se myös luo hajakapasitansseista ja muista parasiittisista komponenteista arviot ja rakentaa niistä piirikaaviomallin. Nämä tiedot ovat hyvin hyödyllisiä esimerkiksi simuloitaessa korkean taajuuden alueella, koska siellä induktiivinen ja kapasitiivinen kytkeytyminen voivat aiheuttaa isoja ongelmia. Ekstraktoitua piirikaaviota voidaankin käyttää, kun halutaan simuloida rakenteiden vaikutuksia komponenttien toimintaan.

5.6 Toiminnallisuuden varmentaminen ekstraktoitujen rakenteiden avulla

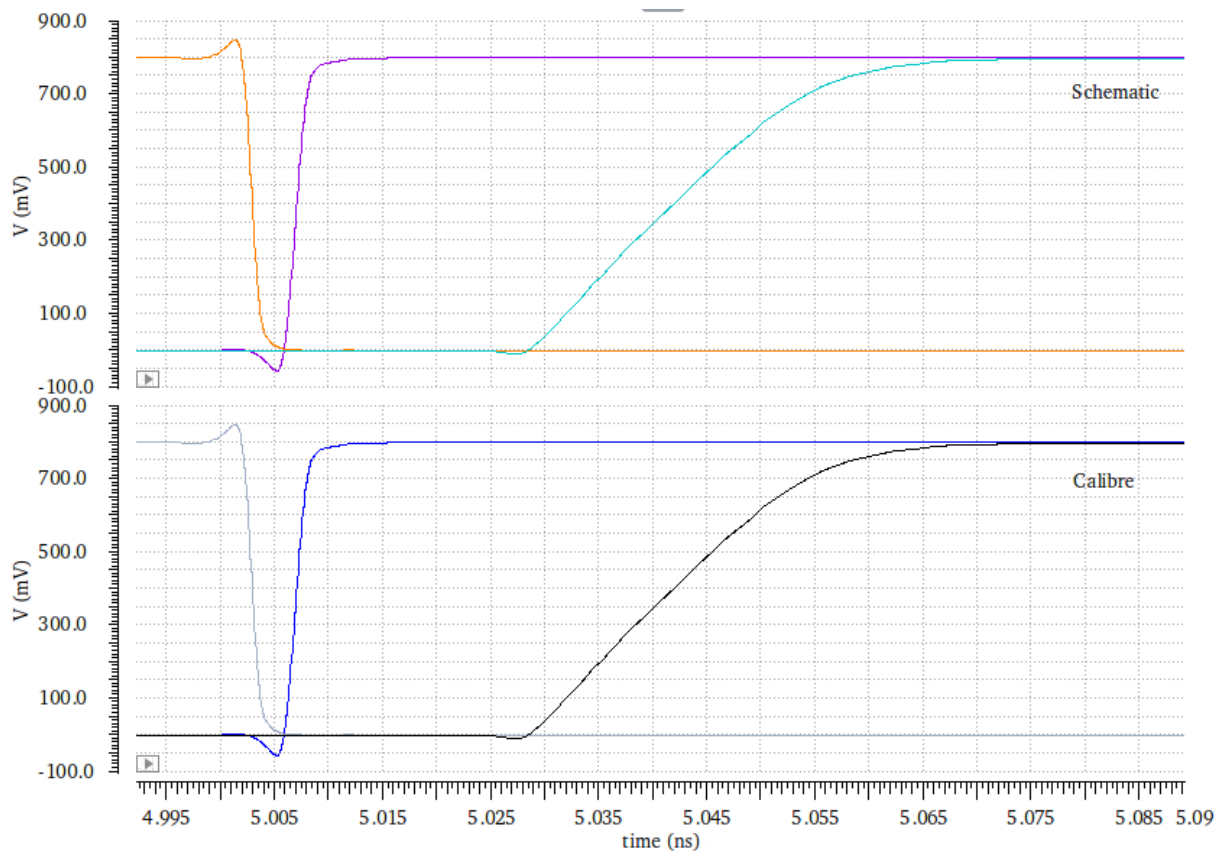
Cadencen normaalien simulointien lisäksi tehtiin kriittisille komponenteille myös varmentavat simuloinnit, joissa käytettiin piirikaaviomallina parasiittiset komponentit omaavaa rakennetta, joka on muodostettu piirikuviosta. Tätä rakennetta simuloimalla nähtiin, miten käytetyt johtimet, pinta-alat ja komponentit toimivat todellisuudessa.

Yksi komponentti, jolle varmentaminen oli hyödyllistä, on virta-DAC. Tämä siitä syystä, että komponentin virrat ovat kohtuullisen suuria (yli 400 μA) ja siihen on sisällytetty myös isohko suodatuskondensaattori. Pitkistä ja liian kapeista johtimista voi aiheutua ongelmia esimerkiksi virrankestävyydessä ja jännitteissä. Kuvassa 45 on esitetty virta-DAC ohjausjännitteen simulointi piirikaavion ja hajakomponentit sisältävän calibre-mallin kanssa. Käytännössä käytetyt taajuudet ovat niin matalia tässä simuloinnissa, ja johtimien rakenne ei aiheuta ongelmia, joten simulointitulokset molemmille malleille ovat identtiset.

Kuvassa 46 nähdään piirikuviosta ekstraktoidun mallin simuloiminen kiikun tapauksessa. Kuvan simuloinnissa on esitetty pito-ajan mittaus, josta kumpikin rakenne antavat tismalleen saman vasteen, joten piirikuvioiden hajasuureet eivät vaikuta oleellisesti myöskään kiikun toimintaan.



Kuva 45. Virta DAC:n lähtöjännitekäyrät. Piirikaavio vs. ekstraktoitu malli (calibre).

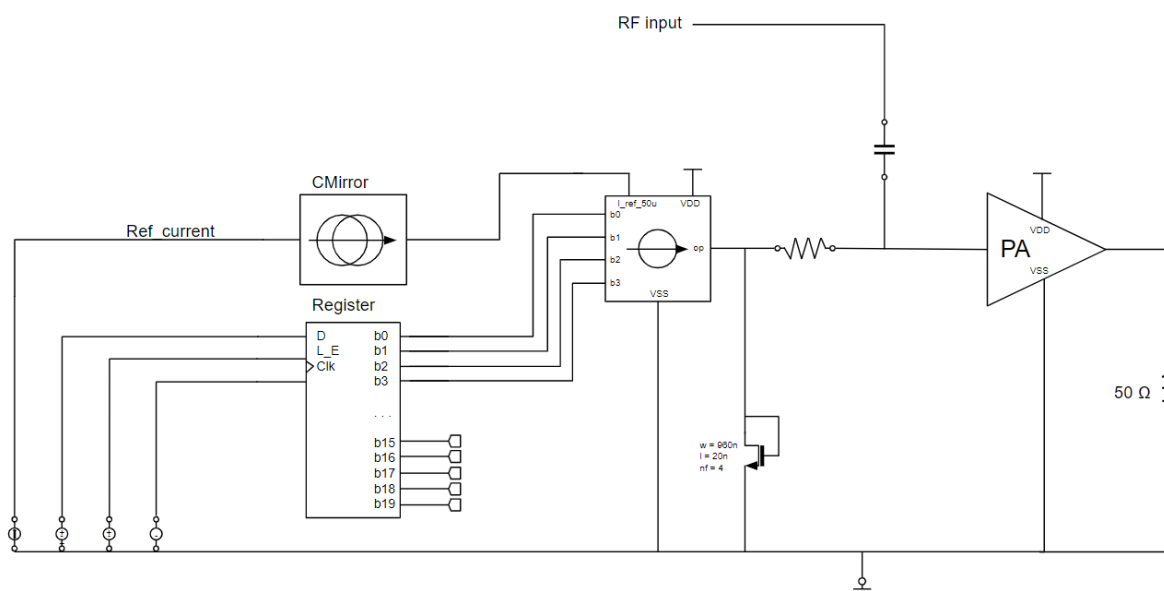


Kuva 46. D-Kiikun piirikaavio vs. ekstraktoitu malli (calibre), pito-ajan simulointitulokset.

5.7 Piirirakenteiden simulointi varsinaisten lohkojen kanssa

Simulointien ja piirikuvioiden jälkeen rakenteiden toiminta varmistettiin vielä varsinaisten lohkojen kanssa erikseen. Rakennettuja digitaalilohkoja otettiin käyttöön RFIC-komponentin ohjaustilanteessa ja ajettiin testipenkkiin kuuluvat simuloinnit. Tämä tehtiin käyttäen ensin piirikaaviotason mallia ja lopuksi vielä käyttämällä ekstraktoitua piirikuviomallia, jolloin saatiin taajuusalueen yli toimivuuden varmennus. Tässä kappaleessa käsitellyssä rakenteessa valmiita logiikkakomponentteja ovat siirtorekisteri, referenssivirtavahvistin ja virta-DAC. Niiden avulla tuotettiin RF-tehovahvistimen biasten ohjaukset. Siirtorekisteri sisältää kirjastoon rakennettuja komponentteja, kuten schmitt-triggerin, D-kiikun, D-salvan ja puskurin, joten käytetty testipenkki antaa kirjastokomponenttien toiminnasta melko laajan käsityksen. Seuraavissa simuloinneissa kuvataan tilannetta, jossa suunniteltuja RF-tehovahvistimen ominaisuuksia muutettiin virta DAC:a säätämällä, ja varsinaiset simuloinnit on tehty RF-simulointeina eri asetuksin.

Kuvassa 47 on esitetty yksinkertainen piirikaavio käytetystä testipenkistä ja ohjauslogiikasta. Kuvan siirtorekisteriin syötetään haluttu data sisään ja kyseinen data otetaan rinnakkaisena ulos. Rekisterin lähdöstä 4 ensimmäistä bittiä ajetaan virta-DAC:in bittituloihin ja niillä kytketään lähdöstä tietyt virtapeilit päälle. Generaattorin luoma virta ajetaan referenssivirtavahvistimen läpi ja sillä ohjataan virta-DAC:in tulovirtaa. Virtageneraattori luo $50\text{ }\mu\text{A}$ suuruisen virran ja referenssivirtavahvistin antaa lähtöön $50,02\text{ }\mu\text{A}$ virran. DAC lähdössä oleva jännite määräytyy diodikytketyn transistorin avulla ja kytketään tehovahvistimen tuloon suuren vastuksen kautta. Kondensaattori erottaa RF-tulosignaalin DC-ohjausjännitteestä, jolla ohjataan tehovahvistimen hilajännitettä. Kuvan testipenkille ajettiin S-parametrisimuloinnit (sirontaparametrit) ja HB-simulointi (Harmonic Balance).



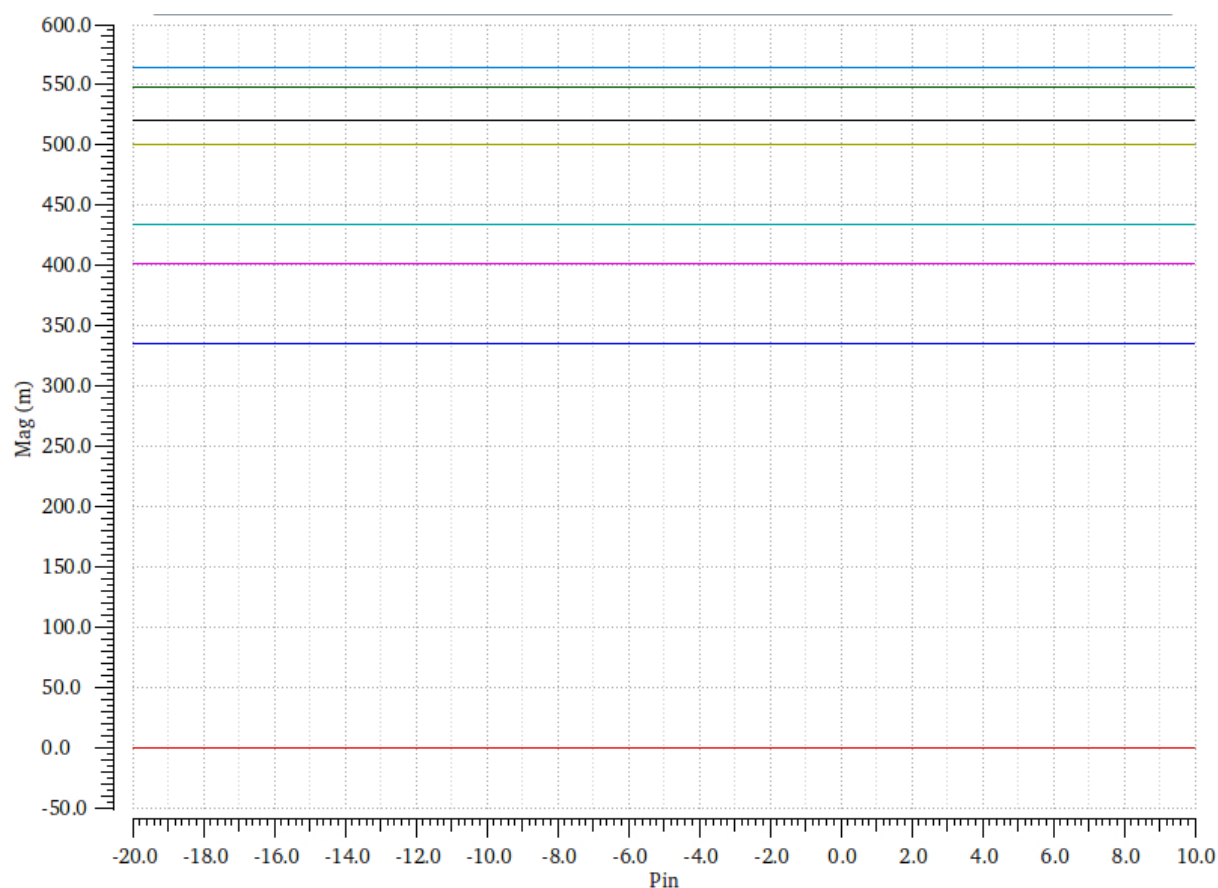
Kuva 47. Testipenkki digitaalilohkojen hyödyntämiseen tehovahvistimen biaksen ohjauksessa.

HB-simuloinnissa piiri on asettuneessa tilassa (steady-state) ja tätä varten luotiin rekisterille verillog-A -malli, joka toteuttaa rekisterin toiminnallisuuden eli siirtää tulodata lähtöön. Näin vältetään pitkältä transienttianalyysin simuloinnilta. Verilog-A -mallissa tulodata syötettiin

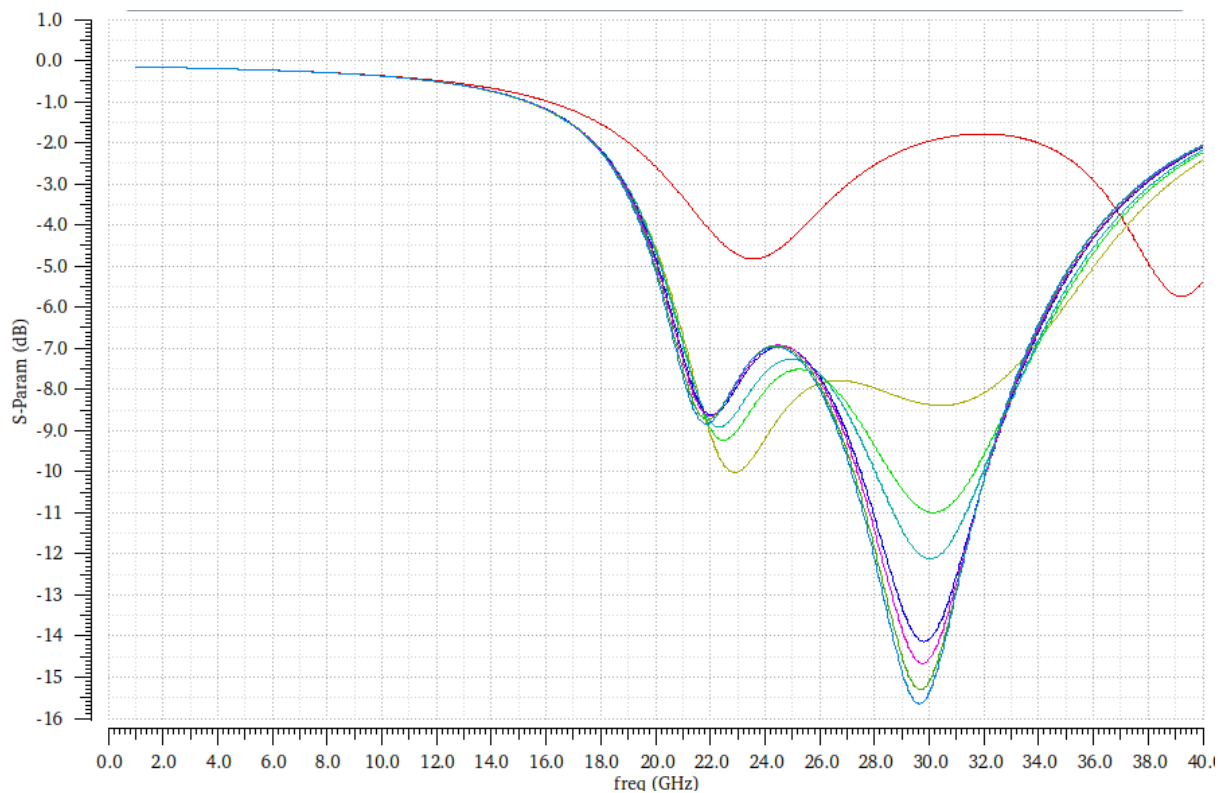
komponenttiin parametrinä ja syötetty data näkyi lähtöpinneissä ilman viivettä. Verilog-koodi toi myös omia haasteitaan simulointeihin. Esimerkiksi koodin muuttujien arvot täytyi päivittää jokaisella suorituskerralla, sillä muuten simulaattori luokitteli ne piilotetuiksi tilamuuttujiksi ja simulointi keskeytettiin. Lopulta simulointien avulla saatiin osoitettua, että rakennettuja logiikkapiirejä voidaan käyttää digitaaliohjaukseen. Kyseinen Verilog-koodi on esitetty Liitteessä 1.

Kuvassa 48 nähdään virta-DAC:n lähtöjännitteet eri bittiarvoilla HB-pyyhkäisy simuloinnissa. Komponentin antamat lähtöarvot eivät juurikaan muutu tehovahvistimen tulotehon muuttuessa. Kuvissa 49-51 nähdään tehovahvistimen S11, S22 ja S21 parametrit pyyhkäistynä useilla eri biasjännitteillä. Punainen käyrä kuvastaa tilannetta, jossa biasjännite on lähellä 0 V ja siitä eteenpäin ohjaus jännitteen kasvaessa käyrät alkavat lähentyä toisiaan. Kuvissa 52 ja 53 nähdään vielä tehovahvistimen vahvistus sekä lähtöteho eri biasarvoilla simuloituna. Kuvista voidaan nähdä, että tehotasojen kasvaessa tehovahvistimen lähtö kompressoituu epälineaarisuuksista johtuen ja samalla biasjännitteen vaikutus pienenee.

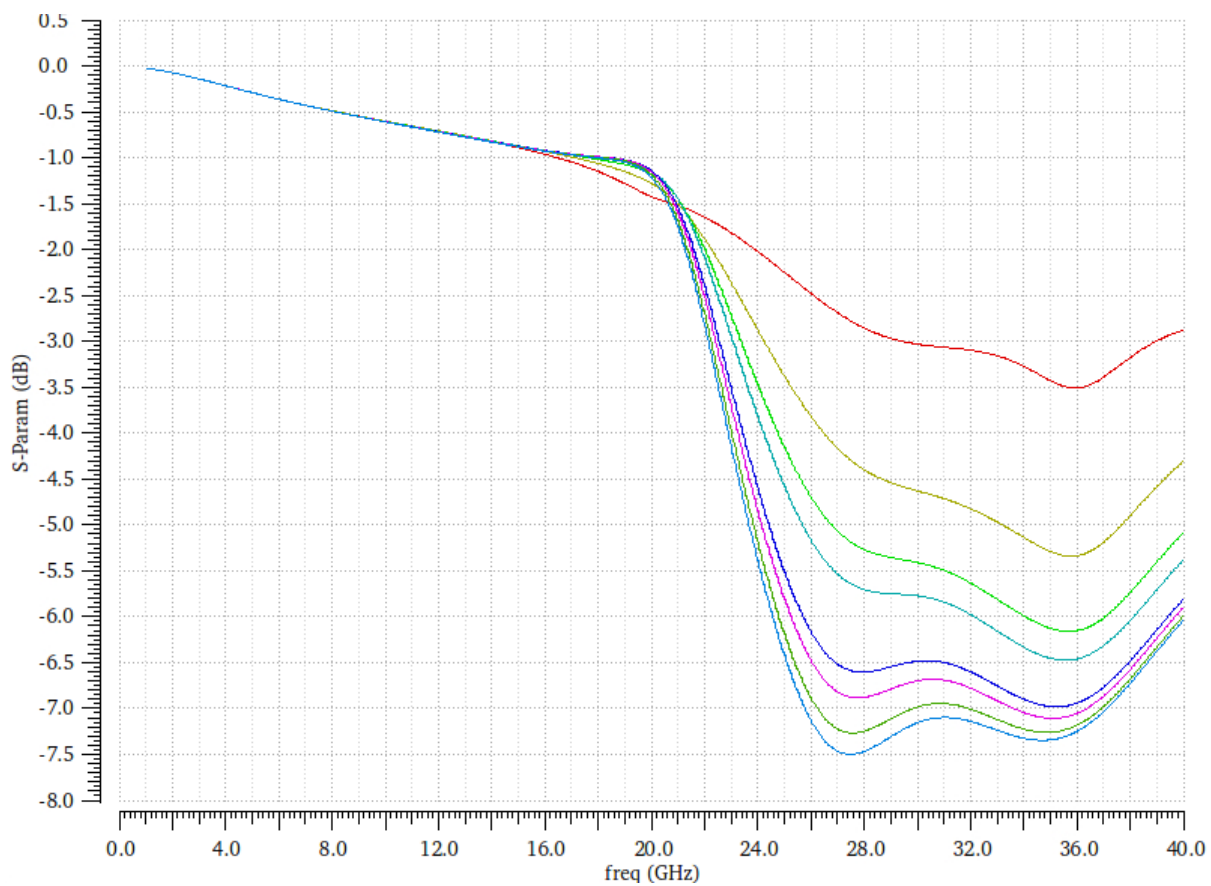
Kuvien perusteella voidaan todeta, että virta-DAC toimii oikein ja sen tuottamaan ohjausjännitteeseen eivät vaikuta käyttötaajuudet eikä ajettavan laitteen teho. Ohjausjännitteen ollessa matala, vahvistin ei toimi, mutta jännitteen noustessa kohti kynnysjännitettä ja sen yli, vahvistin rupeaa toimimaan normaalisti.



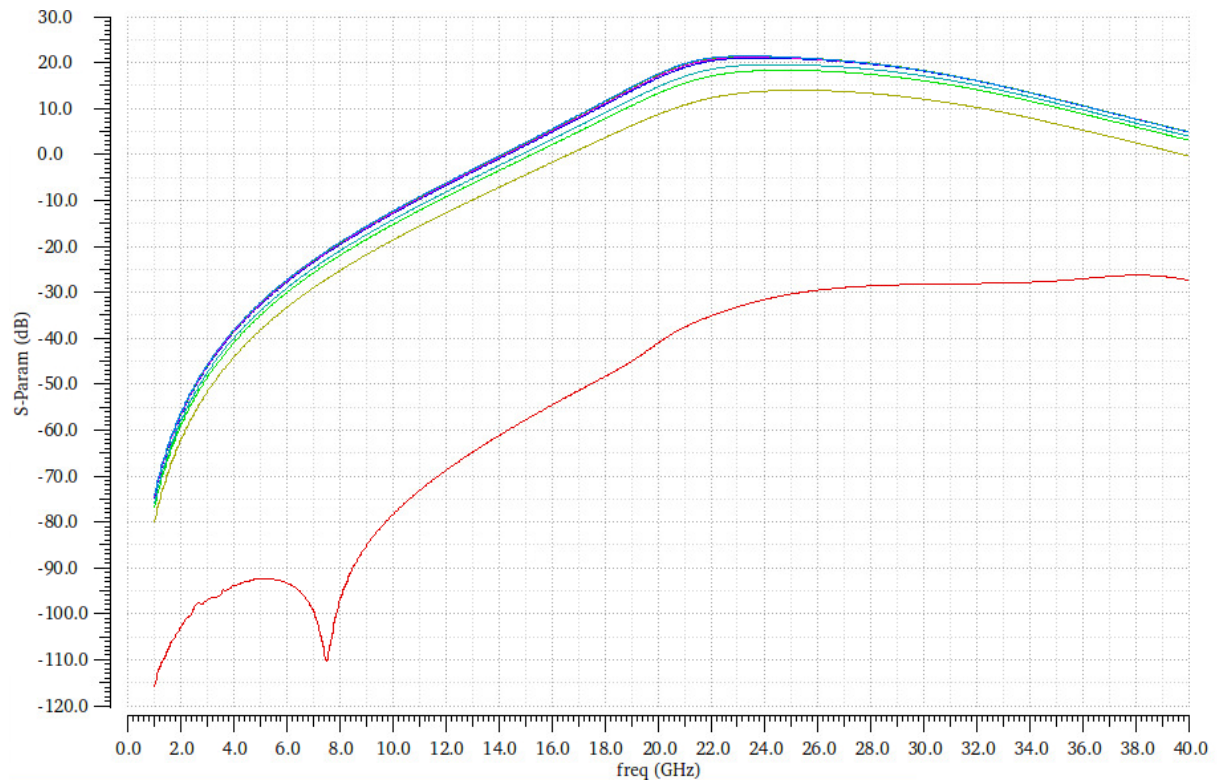
Kuva 48. Virta-DAC ohjatut biakset tulojännitteen funktiona HB-pyyhkäisy.



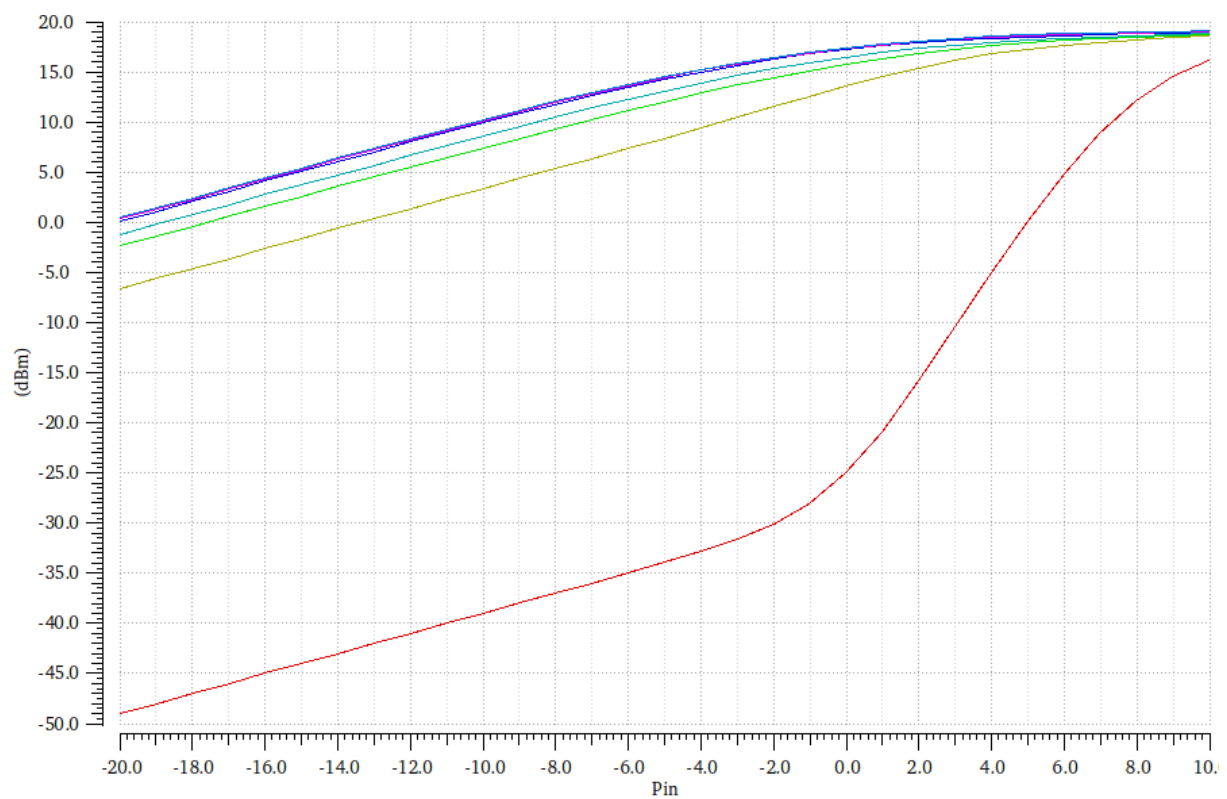
Kuva 49. S11-parametrit eri virta-DAC biasjännitteillä.



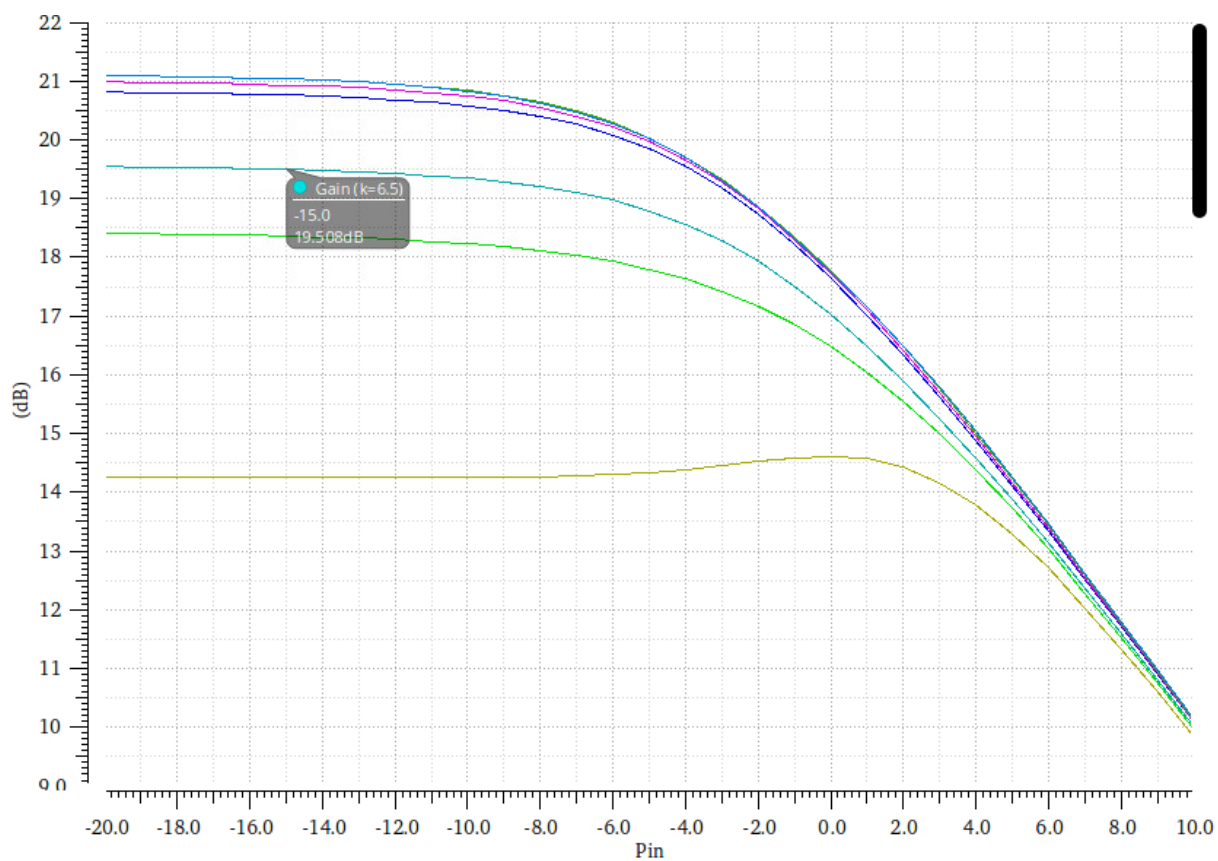
Kuva 50. S22-parametrit eri virta-DAC biasjännitteillä.



Kuva 51. S₂₁-parametreit eri virta-DAC biasjännitteillä.



Kuva 52. Tehovahvistimen lähtöteho eri virta-DAC biasjännitteillä.



Kuva 53. Tehovahvistimen vahvistus eri virta-DAC biasjännitteillä.

6 POHDINTA

Tässä työssä saatiin rakennettua toimiva kokoelma komponentteja analogia- ja RF-piirien digitaaliseen ohjaukseen. Piirien mitoittaminen onnistui sikäli, että piireistä saatiin pienikokoisia ja niiden tehon- ja virrankulutus olivat siedettävällä tasolla. Työssä toteutettiin ainoastaan yhden nopeus- ja vuotovirtaluokan komponenttikokoelma ilman laajempia PVT-simulointeja käyttäen matalan kynnysjännitteen transistorimalleja. Prosessinurkat ja lämpötilanvaihtelut otettiin huomioon vain kriittisimmissä komponenteissa ja varmistettiin, ettei niiden toiminta häiriinny ratkaisevasti eri olosuhteissa. Suunnittelun yksi tavoitteista oli luoda komponentteja, joiden lähdön nousu- ja laskuajat ovat mahdollisimman symmetrisiä, ja tässä onnistuttiin.

Tässä työssä suunniteltiin vain muutama komponentti välittömään tarpeeseen. Tyypillisessä kirjaston suunnitteluprojektissa on lisäksi paljon vaiheita, joihin tässä työssä ei kajottu. Näitä ovat esimerkiksi kirjaston kattavuus, porttien viivemallien generointi ja kirjaston valmistelu logiikkasynteesiä varten.

Kattavan kirjaston luomiseksi täytyisi huolehtia, että sieltä löytyisi monelle kynnysjännitemallille ja ajokyvylle olevat komponentit. Tällöin pystyttäisiin optimoimaan esimerkiksi piirien tehonkulutusta paljon paremmin rakentamalla usean kynnysjännitteen komponentteja, joiden staattinen tehonkulutus on pienempi. Tehonkulutuksen optimoinnissa voisi käytetyssä SOI-prosessissa hyödyntää myös substraatin biasoimista. Sen avulla pystyisi vaikuttamaan aktiivisesti komponentin kynnysjännitteeseen ja vuotovirtaan sekä luomaan adaptiivisia komponentteja, jotka toimivat muuttuvissa olosuhteissa optimaalisesti. Eri ajokyvyn mallit mahdollistaisivat saman komponentin hyödyntämistä erilaisten kuormitusten edessä. Näiden lisäksi komponenteille kannattaisi suunnitella viivemallit, joista pystyisi arvioimaan viiveitä tulosignaalin nousuajan, käytetyn prosessinurkan ja kapasitiivisen kuorman funktiona. PVT-vaihteluiden huomiointi komponenttisuunnittelussa olisi myös hyvä laajentaa kattamaan muitakin nurkkia, kuten FS- ja SF-nurkat, joiden merkitys korostuu, kun käytetään resistiivisiä ylös- ja alaspäin suuntautuvia verkkoja.

Piirikaavioiden mitoittamista ja simulointeja voisi monella tavoin optimoida. Kaikissa tilanteissa paras tapa ei ole käyttää ohjelman tarjoamia jännitelähteitä tai komponentteja simulointien ohjaamiseen, vaan ohjausta voisi suorittaa esimerkiksi skriptien tai verillog-mallien avulla. Verilog-malli mahdollistaa sen, että esimerkiksi komponentin funktionaalisuus ja viiveominaisuudet saadaan toteutettua ilman komponentin sisäiseen rakenteeseen ja solmupisteisiin puuttumista. Riittää, että komponentin toiminta on varmistettu omalla simuloinnilla. Tämä nopeuttaa simulointeja esimerkiksi tilanteissa, joissa halutaan arvioida jonkun komponentin parasiittisia ominaisuuksia ja tuodaan sille signaalit verillog-mallien avulla. Näin simuloitavan piirin solmupisteiden määrä pienenee ja simulointi nopeutuu huomattavasti. Verilog-koodin avulla voisi myös luoda monimutkaisia herätteitä, jotka ovat esimerkiksi normaaleilla jännitelähteillä hankalia toteuttaa. Herätteitä voisi myös koodissa ohjelmoida vaihtumaan kesken simulointien siten, että saadaan simuloitua useilla taajuuksilla tai erilaisilla bittisanoilla yhdessä simuloinnissa.

Testipenkeissä skriptien hyödyntäminen voisi nopeuttaa mittauksia ja simulointeja siten, että skriptin avulla voitaisiin ajaa useita peräkkäisiä simulointeja, vaihtaa komponenttiarvoja ja suorittaa jopa tarvittavat mittaukset sekä kirjata tulokset omaan tiedostoon. Tämä ominaisuus olisi hyödyllinen tilanteissa, joissa simuloidaan ajoitusta, kuten esimerkiksi kiikun setup-ajan mittaaminen. Siinä tulosignaalin viivettä muutettiin alle pikosekunnin verran kerrallaan ja haettiin tiettyä signaaleiden aikaeroa, jolla komponentti ei enää toimi oikein. Myös transistorien sijoittelussa ja johdotuksissa saattaisi olla paljon mahdollisuuksia skripteille. Esimerkiksi

kaikkien komponenttien piirikuvion voisi aluksi ajaa skriptin avulla siten, että ohjelma automaattisesti sijoittelisi transistorit p- ja n-tyypin mukaan sekä piirtäisi käyttöjännitekiskot. Isommissa rakenteissa, joissa käytetään valmiiden komponenttien piirikuvioita, voisi puolestaan hyödyntää samantapaista skriptiä, joka sijoittelisi lohkot ja yhdistelisi tarvittavat pinnit tietyn metallikerroksen yläpuolella. Toisin sanoen kirjastokomponentit olisivat sopivia kytkettäväksi automaattisen johdotustyökalun avulla.

Mikäli toteuttaisin vastaavanlaisen kokoelman suunnittelun uudestaan, pyrkisin luomaan suunnitteluprosessista sellaisen, että kaikkien komponenttien ja lohkojen tekeminen sujuisi samalla tavalla. Prosessien, kuten esimerkiksi komponenttien sijoittelu ja pinnien sijoittelu piirikuvioihin, automatisointi vähentäisi mahdollisesti myös virheiden määrää ja sitä kautta nopeuttaisi rakenteiden tekemistä. Nyt tehdyssä työssä ei ollut kovin kattavia vaatimuksia ja suunnittelussa hypättiin uuteen asiaan ehkä liiankin nopealla tahdilla. Parempi perehtyminen työhön liittyvään teoriaan, suunnitteluympäristöön ja sen käyttöön toisivat suunnitteluprosessiin lisää järjestelmällisyyttä ja ryhtiä. Kuitenkin piirien suunnittelu ja rakenteiden piirtäminen tapahtuu suunnittelijan omalla tyyllillä, ja taidot hioutuvat paremmiksi prosessin aikana, joten oppiminen onnistuu hyvin myös tekemisen kautta.

7 YHTEENVETO

Tässä työssä rakennettiin digitaalilogiikkaa analogia- ja RFIC-komponenttien ohjausta varten. Tavoitteena oli saada aikaan tarvittavat kirjastokomponentit kuitenkin kokonaista digitaalkirjastoa luomatta. Työssä suunniteltiin peruskomponentteja, kuten NOT-, NAND- ja NOR-portteja ja kiikkuja, sekä muutamia monimutkaisempia rakenteita, kuten schmitt-triggeri, virta-DAC ja siirtorekisteri. Komponenttien suunnittelussa käytettävien transistorien valinta tehtiin sen perusteella, mitä tavoitteita piireille oli asetettu. Nopeus ja pieni koko sekä tehonkulutus määrittelivät sen, että päädyttiin käyttämään SLVT-tyypin transistoreja, jotka ovat mahdollisimman pieniä ja nopeita. Työssä rakennettiin komponenttikirjasto vain yhdelle kynnsjännitemallille. Myös piirien prosessi-, lämpötila ja käyttöjännitevaihtelut jätettiin myös vähemmälle huomiolle.

Kirjastokomponenttien suunnittelussa varsinaista digitaalilogiikkaa ei tarvinnut keksiä uudestaan, koska rakenteet ovat yleisesti tunnettuja ja laajalti käytettyjä, mutta jokaisen komponentin mitoitus täytyi tehdä erikseen. Mitoitusprosessi ja simulointi kulkivat aika lailla käsi kädessä. Ensin luotiin piirirakenne, johon arvioitiin komponenttien kokoluokat haluttujen ominaisuuksien perusteella. Tämän jälkeen rakennettiin piirikaavioille sopiva testipenkki, johon luotiin mahdollisimman todenmukainen toimintaympäristö. Porttien simuloinneissa käytettiin kuormana 5 kpl minimikokoista invertteriä ja tulosignaali ajettiin joko invertterin tai viivästysvastuksen läpi, jotta niihin saatiin oikea muoto ja todellista vastaavat nousu- ja laskuajat. Simuloinnissa katsottiin vastaavatko lähtösignaalit haluttua toimintaa ja muotoa sekä tehtiin rakenteisiin tarvittavat uudelleenmitoitukset. Tämä sykli toistui, kunnes saatiin haluttu vaste.

Piirikuvioiden suunnittelu tehtiin siten, että rakenteista saatiin yhteneviä ja helppokäyttöisiä. Jokaisen komponentin korkeus oli sama ja sekä leveys, että korkeus noudattivat kolmannen metallikerroksen jakoa. Leveys määräytyi transistorien lukumäärän ja komponentin kompleksisuuden perusteella. Piirikuvioiden johdotukset pyrittiin tekemään minimiväleillä käyttäen mahdollisimman lyhyitä johtimia. Transistorirakenteita yhdisteltiin jakamalla yhteisiä solmupisteitä niin paljon kuin mahdollista. Tämän avulla saatiin pinta-alaa pienennettyä parhaassa tapauksessa yli 50%. Pinnit nostettiin kolmannelle metallikerrokselle siten, että ne noudattivat metallikerrokselle tyypillistä jakoa, jolloin niiden kytkeminen muualle on mahdollisimman helppoa. Piirikuvioiden ajettiin tarvittavat DRC- ja LVS-tarkistukset, joilla varmistettiin niiden vastaavuus piirikaavioihin nähden. Lopuksi piirikuvioiden vielä ekstraktoitiin parasiiittiset komponentit käyttäen xACT-ohjelmistoa, jolloin saatiin simuloitua piirit myös hajasuureiden kanssa ja testattua niiden toimivuus korkeilla taajuuksilla.

Rakennettujen komponenttien virrankulutus ja tehonkulutus pysyivät halutulla tasolla. Komponenttien keskimääräinen dynaaminen virrankulutus on noin 200 μ A ja tehonkulutus noin 2 μ W/MHz. Viimeinen vaihe työssä oli simuloida rakennetut digitaalilohkot varsinaisten RFIC-lohkojen kanssa ja testata, pystyykö niillä toteuttamaan tarvittavaa ohjausta. Tehovahvistimen hilan jänniteohjaus erilaisissa RF-simuloinneissa toteutettiin käyttämällä referenssivirtavahvistinta, siirtorekisteriä ja virta-DAC:ia. HB-simulointia varten luotiin siirtorekisteristä verillog-A -malli. Lopuksi komponenttikokoelma todettiin toimivaksi ja se otettiin käyttöön.

8 LÄHDELUETTELO

- [1] M. Alioto, "Understanding DC Behavior of Subthreshold CMOS Logic Through Closed-Form Analysis," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1597-1607, July 2010, doi: 10.1109/TCSI.2009.2034233.
- [2] M. M. Mano, M. D. Ciletti, "Digital Design", 4th Edition, Pearson Education, Inc., 2007.
- [3] J. M. Rabaey., A. Chandrakasan, B. Nikolić, "Digital Integrated Circuits, Design Perspective", 2nd edition, Pearson Education, Inc., 2003.
- [4] R. V. Anugraha, D. S. Durga and R. Avudaiammam, "Design and performance analysis of 2:1 multiplexer using multiple logic families at 180 nm technology," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 1849-1853, doi: 10.1109/RTEICT.2017.8256918.
- [5] Yuan Taur et al., "CMOS scaling into the nanometer regime," in *Proceedings of the IEEE*, vol. 85, no. 4, pp. 486-504, April 1997, doi: 10.1109/5.573737.
- [6] K. Roy, S. Mukhopadhyay and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," in *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305-327, Feb. 2003, doi: 10.1109/JPROC.2002.808156.
- [7] A. C. Deng and Y. C. Shiau, "Generic linear RC network model for digital CMOS circuits," *IEEE International Symposium on Circuits and Systems*, Portland, OR, 1989, pp. 860-863 vol.2, doi: 10.1109/ISCAS.1989.100487.
- [8] VEENDRICK, H.J.M. Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits. *IEEE Journal of Solid State Circuits*, New York, v.SC-19, n.4, p. 468-473, Aug. 1984.
- [9] AGARWAL, A. et al. Leakage Power Analysis and Reduction for Nanoscale Circuits. *IEEE Micro*, Los Alamitos, v.26, n.2, p. 68-80, Mar. 2006
- [10] S. Borkar, t. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter Variation and Impact on Circuits and Microarchitecture", *Proc. of Design Automation Conf.*, pp. 338-342, June 2003.
- [11] Yan Han, Jiahua Chen and Jian Liang, "Summary of HV power ICs protecting circuit design," 2001 6th International Conference on Solid-State and Integrated Circuit Technology. *Proceedings* (Cat. No.01EX443), Shanghai, China, 2001, pp. 135-138 vol.1, doi: 10.1109/ICSICT.2001.981441.
- [12] A. L. R. Rosa, L. B. Soares, K. H. Stangherlin and S. Bampi, "Designing CMOS for near-threshold minimum-energy operation and extremely wide V-F scaling," 2015 28th Symposium on Integrated Circuits and Systems Design (SBCCI), Salvador, 2015, pp. 1-6.
- [13] Anantha Chandrakasan; William J. Bowhill; Frank Fox, "Models of Process Variations in Device and Interconnect," in *Design of High-Performance Microprocessor Circuits*, IEEE, 2001, pp.98-115, doi: 10.1109/9780470544365.ch6.
- [14] P. Oßmann, J. Fuhrmann, J. Moreira, H. Pretl and A. Springer, "A circuit technique to compensate PVT variations in a 28 nm CMOS cascode power amplifier," 2015 German Microwave Conference, Nuremberg, 2015, pp. 131-134, doi: 10.1109/GEMIC.2015.7107770.
- [15] W. Yang, Y. Lin and Y. Lo, "Analysis and design considerations of static CMOS logics under process, voltage and temperature variation in 90nm CMOS process," 2014

- International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, 2014, pp. 1653-1656, doi: 10.1109/InfoSEEE.2014.6946202.
- [16] Hongchin Lin and Dern-Koan Chang, "A Low-Voltage Process Corner Insensitive Subthreshold CMOS Voltage Reference Circuit," 2006 IEEE International Conference on IC Design and Technology, Padova, 2006, pp. 1-4, doi: 10.1109/ICICDT.2006.220833.
- [17] L. A. Glasser, D. W. Dopperpuhl, "The Design And Analysis of VLSI Circuits", Addison-Wesley Publishing Company, 1985.
- [18] B. H. Calhoun et al., "Digital Circuit Design Challenges and Opportunities in the Era of Nanoscale CMOS," in Proceedings of the IEEE, vol. 96, no. 2, pp. 343-365, Feb. 2008, doi: 10.1109/JPROC.2007.911072.
- [19] Zabeli, Milaim & Caka, Nebi & Limani, Myzafere & Kabashi, Qamil. (2007). The impact of MOSFET's physical parameters on its threshold voltage.
- [20] C. Xu, X. Wang and W. Liu, "Threshold Voltage Tuning Of 22 nm FD-SOI Devices Fabricated With Metal Gate Last Process," 2019 International Conference on IC Design and Technology (ICICDT), SUZHOU, China, 2019, pp. 1-4, doi: 10.1109/ICICDT.2019.8790860.
- [21] C. XU, C. WANG and D. LIAO, "Advanced 22nm FD-SOI Technolgy With Metal Gate Last Process," 2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC), Xi'an, China, 2019, pp. 1-3, doi: 10.1109/EDSSC.2019.8754434.
- [22] P. Flatresse, "Process and design solutions for exploiting FD-SOI technology towards energy efficient SOCs," 2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), La Jolla, CA, 2014, pp. 127-130, doi: 10.1145/2627369.2631640.
- [23] S. N. Ong et al., "22nm FD-SOI Technology with Back-biasing Capability Offers Excellent Performance for Enabling Efficient, Ultra-low Power Analog and RF/Millimeter-Wave Designs," 2019 IEEE Radio Frequency Integrated Circuits Symposium (RFIC), Boston, MA, USA, 2019, pp. 323-326, doi: 10.1109/RFIC.2019.8701768.
- [24] T. Maekawa, S. Amakawa, N. Ishihara and K. Masu, "Design of CMOS inverter-based output buffers adapting the cherry-hooper broadbanding technique," 2009 European Conference on Circuit Theory and Design, Antalya, 2009, pp. 511-514, doi: 10.1109/ECCTD.2009.5275025.
- [25] N. Balabanian, B. Carlson, "Digital Logic Design Principles", John Wiley & Sons, Inc., 2001.
- [26] K. J. Breeding, "Digital Design Fundamentals", 2nd Edition, Prentise-Hall, Inc., 1992.
- [27] B. N. Bagamma, K. S. V. Patel and P. Ravi, "Implementation of 5–32 address decoders for SRAM memory in 180nm technology," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, 2017, pp. 110-114, doi: 10.1109/ICEECCOT.2017.8284649.
- [28] S. Alahdab, A. Mäntyniemi and J. Kostamovaara, "A 12-bit digital-to-time converter (DTC) with sub-ps-level resolution using current DAC and differential switch for time-to-digital converter (TDC)," 2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Graz, 2012, pp. 2668-2671, doi: 10.1109/I2MTC.2012.6229685.

9 LIITTELUETTELO

Liite 1 20-bittisen siirtorekisterin verilog-A koodi


```

        else if (str == "1")
            Data_in = (Data_in^m_1);
        else
            Data_in = (Data_in^m_0);
        if (i!=0)
            Data_in = Data_in << 4;
        end

//Set the output values
        if ((Data_in & mask0) == 1)
            V(b0) <+ V(VDD);
        else
            V(b0) <+ V(VSS);
        if ((Data_in & mask1) == 2)
            V(b1) <+ V(VDD);
        else
            V(b1) <+ V(VSS);
        if ((Data_in & mask2) == 4)
            V(b2) <+ V(VDD);
        else
            V(b2) <+ V(VSS);
        if ((Data_in & mask3) == 8)
            V(b3) <+ V(VDD);
        else
            V(b3) <+ V(VSS);

        ...

        if ((Data_in & mask19) == 524288)
            V(b19) <+ V(VDD);
        else
            V(b19) <+ V(VSS);
    end
endmodule

```